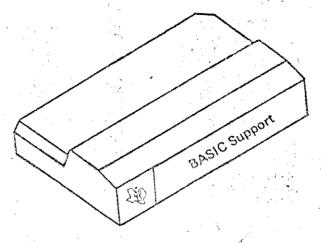


from PCIF



# TO YOUR 99/4

BLOCK MOVES, CHAINING, CRU ACCESS, 2K BYTES CPU RAM FOR MACHINE CODE SUBROUTINES, AND MUCH, MUCH, MORE



# TI BASIC Support Module Pricing

with 2K RAM - \$150.00 without 2K RAM - \$100.00

Available to TI Cost Centers and individual TIer's and their families.

To order call Sally Hayes in Dallas, 995-7131. Allow two weeks for delivery.

# BASIC SUPPORT LIBRARY USER'S GUIDE ( PRELIMINARY) INTRODUCTION

In order to form a more perfect union between the TI 99/4 and the BASIC programming language, this library provides access to various memory resources via subroutine calls from BASIC. These memory resources include VDP memory, CPU memory, CRU interface registers, VDP registers, and disk sectors.

The names of these support subroutines each begin with the character "@" in order to distinguish them as belonging to this collection.

Since two modules won't fit in one slot, the routines SAY and SPGET are also included for speech synthesizer interface.

### SUMMARY

The BASIC support module was produced to enable an experienced programmer to get at the features of the 99/4 through BASIC. It does not require a memory extension.

Some of the major features are:

BLOCK MOVES: Allows for several screen images to be created and subsequently displayed rapidly. An example of use includes the agriculture demonstration where a screen full of data was quickly displayed rather than BASIC updating the screen line by line. Apparent screen motion can be obtained by rapidly updating the display memory.

EXECUTE 9900 CPU CODE: Allows a BASIC program to execute a very short (48 byte) 9900 program. This feature was used in generating a BASIC program to test the CPU memory extension.

PACKING UTILITIES: Allows for saving memory space. A 99/4 numeric variable takes 8 bytes. These routines allow 64 bits, 8 one-byte integers, or 4 two-byte integers to be stored into one location. This very useful if the program requires a large number of small integers.

LOGICAL ROUTINES: BASIC does not allow logical operations, a frequent technique used by programmers. This will allow one to draw lines and curves and other graphics quickly from BASIC.

CRU ACCESS: Used in VTR to read tape position. Used in detecting telephone rings in TIVUE system. This is generally used for controlling devices and detecting external stimuli.

PROGRAM CHAINING: A facility is provided for saving a program on a peripheral device and subsequently re-starting the program when invoked from another program.

<u>PLAY Sound Table</u>: The programmer may use these routines to construct and play sounds of more than the usual single chord, at maximum speed (60 sounds/second) and in parallel to other program activities.

### BLOCK MOVE

Block move routines are supplied for moving large blocks of data within or between memory resources. The types of memory include VDP, RAM, CPU RAM or ROM, G-ROM.

The general form of a CALL to one of these routines is:

CALL @X2Y (Len, Src, Dest)

where X and Y are each V, C or G. X represents the type of memory from which to move; Y represents the type into which to move.

V indicates VDP RAM

C indicates CPU (9900) memory address

G indicates a GROM

<u>Len</u> represents a variable whose value is the number of bytes to be transferred.

Src represents a variable whose value is the source (from) address.

Dest represents a variable whose value is the destination (to) address.

Note that Len, Src and Dest are all variable names and may not be general expressions.

# VDP Addressing (V)

A variable containing a VDP address may have a value from 0 to 16383 (>3FFF). Moving to or from addresses outside this range can have

# CPU Addressing (C)

CPU (9900) memory has addresses from 0 to 65535 (>FFFF).

# GROM, Addressing (G)

GROM addressing only applies to library module 0. GROM has addresses from 0 to 65535 (>FFFF). GROM as Dest is NOT ALLOWED.

### Example 1:

To move a screen image into CPU memory at > 2000,

L = 768

V = 0

C = 8192

CALL @V2C (L, V, C)

NOTE: The screen image is at VDP RAM location 0 thru 767 (> 2FF).

### Example 2:

To move the screen up 1 line (32 characters)

L = 768 - 32

V1 = 32

V2 = 0

CALL @V2V (L, V1, V2)

# SIC VARIABLE ADDRESS

To acquire the VDP RAM address of a BASIC variable, use the @LOC routine.

CALL @LOC (Var, Addvar)

Var and Addvar represent two BASIC variable names. The VDP RAM address of the first (low address) byte of Var is placed in Addvar.

Example: Save the screen image in array A.

DIM A (95)

CALL @LOC (A(o), AL)

L = 768

V = 0

CALL @V2V (L,V,AL)

NOTE: Each numeric variable requires 8 bytes so that 96 array elements thru 95) are sufficient to hold 768 bytes.

NOTE: CALL @V2V (L,AL,V) will replace the saved screen image.

NOTE: CALL @LOC (AS,AL) will give the address of the string AS in AL, but this value of AL must be used before the string AS is re-allocated by the system. Numeric variables are not re-allocated!

### CRU ACCESS

There are 2 routines for CRU access, one for writing, one for reading.

CALL @CRIN (Addr, Bits, Data)

CALL GCROUT (Adde, Bits, DGTA)

will cause the number of bits specified by <u>Bits</u> to be transferred from (TO)

10 BASIC variable specified by <u>Data</u> into athe <u>CRU</u> at the <u>address</u>
specified by <u>Addr</u>. The bits to be written, should be right justified in <u>DATA</u>. The number of bits should be between 1 and 16.

## Execute 9900 (CPU) Code:

CALL @CEX (RO,R1,R2)

Causes the 48 bytes at VDP RAM location > 7D0 to execute as if it were a 9900 program at CPU RAM location > 8300. This is strictly "user beware" and the 9900 program must have the following form:

- Locations >700, >701 must contain a pointer to the first instruction to be executed.
- If the workspace pointer is changed, then it must be restored to 83EO before returning to BASIC.
- In order to return to BASIC, the last 9900 instruction executed must be:

B \*R11 (>0458)

The register values at > 8300, > 8302, > 8304 will be returned in RO,R1,R2 respectively.

The simplest way to get a program into VDP RAM location >7DO is to use the CHAR function with character codes 154 thru 159.

### Example:

CALL CHAR (154, "83020458")
CALL @CEX (A,B,C)

will execute a return (B \*R11) to BASIC, i.e., a NO-OP:

### EXIT BASIC

To simulate a "SHIFT Q" from your program, execute the statement CALL @EXIT

nothing in memory will be recoverable.

MODIFY VDP REGISTER (Handle with Care)

To put the 1-byte value "Value" into the VDP register "Reg",

CALL @VDRPUT (Reg, Value)

Where Reg and Value are BASIC variables containing the appropriate values.

# Adjust Character Codes

To modify the contents of a string of bytes in VDP RAM by adding a constant to each one,

CALL @VADD(L,Delta,Bytes)

### LOGICAL OPERATIONS

CALL @VAND(L,V1,V2)

The string of bytes beginning at the address contained in V1 will be logically "AND" ed with (and the result stored into) the string of L bytes at the address contained in V2.

CALL @VOR (L,V1,V2)

does the corresponding "OR" operation.

### ONE-BIT FLAGS

To treat a numeric variable as 64 individual 1-bit flags;

CALL @ON(V,B)

to set bit B in variable V to one;

CALL @OFF(V,B)

to set bit B in variable V to zero;

CALL @TEST(V1,B,V2)

to get the value of bit B in variable V1 and put it in V2;

CALL @VON(V,B)
CALL @VOFF(V,B)
CALL @VTEST(V,B)

function the same as @ON; @OFF, @TEST respectively, except that V is a numeric variable whose value is the address of the \$-byte VDP RAM area whose bit is to be set.

 $i_n$ 

#### Packing Bytes

If in any array of 8 numeric variables, each variable has an integer value less than 256, then these 8 values can be packed into a single numeric variable by

CALL @PUTB (A(X),B)

The 8 values in A(X), A(X + 1) thru A(X + 7) will be packed into the variable B, left to right (exponent first).

The reverse operation is accomplished by

CALL @GETB(B,A(X))

### Packing Words

Similar to Packing Byes, except that 4 variables of integer value up to 216-1 (85835) are packed into a single numeric variable.

Pack them in by

CALL @PUTW(A(X),B)

and unpack them by

CALL @GETW(B,A(X))

#### CHAINING

The BASIC SUPPORT MODULE provides a checkpoint and chaining facility. At any time after a BASIC program has been loaded into the TI 99/4, (whether it has been executed or not), the user may "checkpoint" that program onto an external device. This "checkpoint" file may be re-started from another BASIC program (chaining) at some later time. A re-started program will have retained all previously computed values of ariables and all special character graphics for character codes above 95.

### CHECKPOINT

CALL CHECK (VAR\$)

VAR\$ is the name of a string variable whose value is the name of the file onto which to save the current program and associated parameters.

NOTE: The call to CHECK must be an imperative statement, not a program statement.

# RESTART

CALL CHAIN(VAR\$)

VAR\$ is the name of a string variable whose value is the name of the file from which to retrieve a program to be re-started.

NOTE: The call to CHAIN must be a program statement, not an imperative statement.

# PLAY Sound Table

CALL @PLAY(VADD)

VADD is the name of numeric variable whose value is the VDP RAM address at which is located the sound table to be "played". This "playing" will continue until a "stop" code is encountered or until another CALL @PLAY is executed, or CALL Sound, or any system function which causes a "beep".

### CALL @PLABK(VADD)

If this statement is executed immediately after a CALL SOUND statement, on 11-byte string of sound definition and a 6-byte "stop", string will be placed at the VDP RAM location whose address is the value of VADD.

### ACQUIRE SYSTEM DEPENDENT VALUES

The following subroutines may be called to acquire the corresponding values:

@MEM	CURRENT AVAILABLE VDP RAM
@MAXI	MAXIMUM POSITIVE VALUE
0MINI	LARGEST NEGATIVE VALUE
@EPS	SMALLEST POSITIVE NUMBER
@BIGI	LARGEST CONSECUTIVE INTEGER

### Example:

CALL @MEM(A)

will cause the value of A to be changed to the number of bytes remaining available to strings, subroutine stack, etc., at the time of the CALL.

# Display a Line of Text

CALL @AT(R,C,T\$) will display the string value of the variable TS horizontally from left to right on the display screen (with "LF/CR" after column 32) beginning at the row and column indicated by the values of variables R and C respectively.

Specifically,  $\hbox{\it @AT}$  acts exactly like HCHAR, with T\$ substituting for the string of multiple copies of the same character.

# CALL a Known GPL Subroutine

CALL @GSUB(GADDR) will transfer to the GPL subroutine located at the address indicated by the value of the variable GADDR. Some examples of useful subroutines are:

ADDRESS	ROUTINE
7201C	REDEFINE DEFAULT CHARACTER SET
MOIR	STUNG-SPACE GARBAGE COLLECTION



CRU ADDR IS IN 01.Q1+1

10/14/83

14:07:17

PAGE 1

CONTROL OF THE SECOND OF THE SECOND S

GPL ASSEMBLER 79.150

```
14:07:17
                           '85M'
                                       18/14/83
 GPL ASSEMBLER 79.150
                                      18 174
                                                  Q2+1
                               # BITS
             75
                         OUTPUT:
             76
                               CRU PATTERN
                                                  00<6
             7.7
             78
                         MOVE 12 FROM ROM(#CRINDATA) TO 2>C
             79 CRIN02
5050 31000C
5060 00602F
                                             POINTER TO MACHINE LANG CODE
                         DST
                               >8300.20
6053 BF0083
             80
6066 90
                         DAND >000F.802
                                            MASK OFF 12 BITS
a067 830600
             81
aCaa QF
                                             SHIFT # BITS INTO POSITION
                          OSLL 802.6
606B E30600
              82
aCaE Da
                                             COMBINE STOR INSTR & # BITS
606F 651206
                          DOR
                               @02.@>12
              83
                               >FO
4072 OFF0
              2.4
                          3 23 1
                          RIN
8074 00
              85
                          8.5
                         DATA #CEX.#GRO(81
              87 CROUT
6075 609960
A078 80
                          DATA 6.: @CROUT:
5079 054043
              88
6070 524F55
607F 54
                          ENTRY POINT FROM BASIC
              89
                          CALL P
6080 066125
              90
                 CROTG1
                          CALL CROTU2
5083 055089
              71
£066 060012
              92
              93
                          ENTRY POINT FROM G-LANGUAGE
              94
              95
                          IMPUT:
                                CRU ADDR IS IN
                                                  01,01+1
              96
                                                  02+1
                                # BITS IS IN
              97
                                                 03.03+1
                                CRU PATTERN IS IN
              98 -
                  *
                          OUTPUT:
              99
                               NONE
             100
             101
                  CROTO2 DST
                              a01.210B
             102
6089 806802
                         ST
             103
                               aQ2+1.aI0B+2
608C BC6A07
                                BUF.@10B+3
                          ST
608F BE686C
            104
                           DST
                                aQ3,aBUF
6092 BD6CCA
             105
                           1/0
                               alog.3
 5095 F66803
             106
 6098 00
             107
                           RIN
                 108
                           DATA #CEXP.#CEXO1
 6099 60A260.
              109
                 CEX
 6090 DF
                           DATA 4.: &CEX:
 5090 044043
              110
 6040 4558
              111 CEXP
                          DATA #LOC.#CEXPO1
 50A2 518260
 60AS AC
 60A6 054043
                           DATA 5.:@CEXP:
              112
 40A9 495850
 60AC 055125
             113 CEXP01
                          CALL P
                           DST
                                891.828
 ACAF BDGCG2
             114
                                >F/1
 6082 OFF0
              115
                           XML
 6084 8F0040
                          5.1
                                apo .800. O GOTO CEXPO2
              115
 5087 DC
                          09T - 002.2>0
              117
 8088 B00005
```

And the second of the second o

79.

PAGE

48SM1

10/14/83

14:07:17

GPL ASSEMBLER 79.150

```
6157 6167
                             CALL
                                   COMMA2
6159 U5616A
              160
5150 BDBE4A
               151
                             DST
                                    SFAC. 304
615F B01000
               162
                             DST
                                    20.204+2
                                    aCHAT .NE. RPARS GOTO ERRY
                             5 I F
6162 D64286
               153
6165 4600
6167 OF18
                                    PGMCHR
               164 FINIS
                             XML
6169 00
               165
                              RTM
               165
                              PARAMETER FETCH
               157
               168
616A 064283
                    COMMA2
                              $1F
                                    achat .NE. COMMA$ GOTO ERR7
               169
6160 469D
616F 0F1B
                                    PGMCHR
               170
                              XISL
                                    SYM
                    PARM
6171 CF13
               171
                              X 111
5173 OF14
               172
                              XIIL
                                    SMB
                              DST
                                    aFAC++.aG
6175 BDC04E
               173
                             MOVE
5178 350008
               174
                                    8 FROM RAM(@0) TO @FAC
5178 4ABOOC
617E 0555CF
               179
                              CALL
                                    CVELIN
6181 00
               178
                              RTH
6182 619061
               177
                    LOC
                              DATA
                                   #G2V.#LOCU1
春185 春8
6186 04404C
               178
                              DATA
                                   4.: @LOC:
6189 4F43
6189 056125
               179
                              CALL
                    L0001
                                    201+2.80
618E BD9004
               120
                              DST
                              CALL
                                    CIF
6191; 0555A1
               181
                                    8 FROM 8FAC TO RAM(802+2)
6194 350008
               182
                              MOVE
6197 B0084A
               183 .
519A 050012
                              CALL
                                    RPL
                                    #C2V.#G2V01
6190 618361
               184 G2V
                              DATA
61A0 A6
61A1 U44U47
               185
                              DATA
                                    4 : 2627:
61A4 3256
61A6 056125
               185 G2V01
                              CALL
                                    aQ1 FROM ROM(aQ2) TO RAM(aQ3)
61A9 3202B0
               187
                              MOVE
61AC 0A0000
61AF 06
5180 050012
               188
                              CALL
                                    #V2G.#C2V01
6183 61CD61
               189
                     C2V
                              DATA
5185 BC
6187 044043
               190
                              DATA
                                    4.:ac2V:
618A 3256
               191
61BC 056125
                     C2V01
                              CALL
61BF A70483
               192
                              DSUB
                                    CB1AS, 802
6102 00
6103 340280
                              MOVE 801 FROM 80(02) TO RAM(803)
               193
6106 0AC000
6109 05
               194
                              CALL RPL
61CA 050012
SICD 61E361
               195
                     V25
                              DATA
                                    #V20.#V2G01
6100 06
6101 044056
               196
                              DATA 4.: & V2G:
6104 3247
```

'BSM'

GPL ASSEMBLER 79.1EU

14:07:17

10/14/53

PASE

G₽L.	ASSEMBLE	ER 79.	150	189	<b>初</b> 了	10/14/83	14:07:17
5109	056125 200200 000ABC		V2601	CALL MOVE	P ao1 FROM	RAM(@Q2) ]	FO ROM(803)
61E0	060012 61FD61	199 200		CALL DATA		:01	
61E7	044056	201		DATA	4.:aV20:		
61EA 51EC 51EF 51F2	056125 A70A83	202 203	V2C01	CALL DSUB	P CBIAS, 8Q	Š	
61F3	340200 000ARQ	204		MOVE	201 FROM	RAM(802)	<50) <b>05</b> OT
61FA	060012			CALL		0.04	
61FD 5200		28 <b>6</b>	G2C	DATA	#626.#C2	uui	
6201 6204	044043	207		DATA	4.:2020:		
6205	056125		02001				
6209 6200		205		DSUB	CBIAS.ag	T T	
5200	A70A83	210		08UB	CBIAS.@Q	3	
6214	340200 000AC0	211		MOVE	€01 FROM	20(Q2) TO	<b>&amp;</b> U(03)
621C	05006 050012 523352 25		G26	CALL		G01	
6220	044047 3247			DATA	4.:@626:		
5225	055125				p		TA BAH (007)
	2AB200 000A00	216	***	MOVE	aqı FROM	ROM (802)	TO ROM(8Q3)
	6006	en. 4 mg			0.01		in the North American
		217 218		CALL DATA		601	w · ·
				DATA	4.:@026:		
	066125		C2691	CALL	Ρ		*
523F 5242	A70483	221		DSUB	CB1AS.aG	Tr dec	
6243 6246	200200 000AC0	222		MOVE	201 FROM	( 02) TO	) ROM(@Q3)
	0006 050012	223		CALL	RPL		#
524E 5251	626962 57	224	620	DATA	#CRUN.#G	12001	
8252	044047 3243	225			4.:2620:	1	•
	066125 A70A83	225 227	G2C01	CALL DSUB	P CBIAS.80	3	

· · ·

45

 $\epsilon_{\rm a}$ 

\*\*\*\* \*\*\*\*

PAGE

The state of the s	A COLOR DE CONTRACTOR DE C	and the second s
	તું કર્યા કરવામાં અનુસર્વા કર્યા કર્યા છે. તે અને અનુસર્વા એ કર્યો મોટી અને મોટી અને મોટી અને મોટી કર્યો કરો ક	significations was was some more a property of the first of the contract of th
The state of the s	<ol> <li>** ** ** ** * * * * * * * * * * * * *</li></ol>	the second control of

GPL	ASSEMBL	ER 79.	.150	189	SM' 10/14/83 14:07:17 PAGE 6
625D		000		MOUE	ag1 FROM ROM(ag2) TO ag(@3)
	320200			HOVE	AMI LEGGE WOLLGATE LO EDEACE
	000A00			,	
6264		000	-		RPL
	060012				#VDRPUT.#CRUNO1
6267 6260	627862 77	ಎಎಟ	CRUN	DATA	#VDKTU: #UKUHUI
	054043	10 T/A		DATA	5.:@CRUN:
	52554E	24 ಚಿತ್ರ		vers	₩ . • @ \ (\text{Contents} )
		232	CRUNO1	CALL	PGMCTR .
	0F20	233	41101101	XML	•
	0.60012	234		CALL	RPL
	628562	235	VORPHT	DATA	#VADD.#VDRPG1
527E		***			
	074056	235		DATA	7.:2VORPUT:
	445250				
6285	5554				
6287	005083	237	VORDATA	DATA	#>0060.#>8302 MOV >8302,R
528A					
6289	050108	238		DATA	#>05C1.#>0801,#>8CO2 SUP8: MOVE
	018002				
5291	050108	239		DATA	#>06C1.#>D8O1,#>8CO2,#>045B SWPB; MOVB
6294	018002				
5297					
5299	მაგ125	240	VORPO1	CALL	b ,
629¢	E30200	241		DSLL	891.8
629F					
	BC0307				202+1.201+1
	860280				801.7
	310012			MOVE	18 FROM ROM(#VDRDATA) TO 8>4
	046287			4	
	BF0083	245		មូន៖	>8304.20
	04: 0FF0	011		U +41	X 100 40
				XML	
5252	060012	247 248		CALL	The state of the s
4.500	400440			πατα	#VOR,#VADDO1
6288		47.	YAVV	Pain	#*0//, #*/00001
	054056	250		ΠΔΤΔ	5.:aVADD:
	414444			F-11111	J * * C * 11, D C *
			VADD01	CALL	P .
	A1020A	252			aq3.aq1
	010A02				203 .DLT. a01 1
	6203				
	AOBOOA	254		ADD	202+1.RAM(203)
6200					
	910A	255		DINC	a03 1
	056205	256		<b>\$</b> END	
	050012			CALL	RPL

259

250

251

VORU1

6206 62F962

620A 044056 6200 4F52 620F 066125

6282 A1020A

6209 DF

0

\*\*\*

Fee 1

DATA #VAND.#VORO1

DATA 4.: @VOR:

DADD aq3.aq1

CALL P

	σPL	ASSEMBLE	ER 79.	150	· 8:	3t4 4	10/14/83	14:07:17	PAGE	7
		D18A02 62F6	262		EWHILE	803 .DLT.	891			1
ŧ.	2EA	54600A 8004	263		OR	RAM(802).	RAM(@03)			1
		910A	264		DINC	203				1
ع,	2F1	9105	265	1 -	DINC	802				1
4	2F3	056265	266	•	\$EMD					
		050012	267		CALL I	RFL				
€,	2F 0		268	VAND	DATA	#RSECT.#V	ANDO1			
		054056	269		DATA	5.:@VAND:				
		414E44 066125	~~, ~~, ~, ~,	LIANDON	C 0 1 1	0				
		000125 A1020A	270 271	VAN001	CALL Dadd	P 207 201				
		010A02	272			8Q3.AQ1 8Q3 .DLT.	204			1
		631A	Alm I alm		4.34117 117	TWO ADMIL	u.u.i			1.
		B0800A	273	•	AND	RAM(802).	(EQS)			1
		8008								
		910A	274		DINC	<b>२०</b> उ				1
		9105	275		DINC	802				1
		055309	275		#END					
Č	101A	050012	277 278		CALL	RPL				
		0030	279	* STLN	EQU	>30				
		0032	220	ENLN	EQU	>32				
		0000	281	SRC	EOU	0				
		0002	282	DEST	EQU	2	•			
		0070	283	PTOP	EQU	>70				
		2840	284	PRGLST	EQU	>2840			_	
		03EC	285	SAVEPC	EOU.	>3EC	$\hat{z}_{a}$	* .	*	
		0024	285	STVSPT	EQU	>24	A Company of the Comp	* * * .		
		OSF8	287	VRAMVS	EQU	>5F8	-			
		0044	288	BUFFY	EQU	>44				
		0034	289	DATA	EQU	>34	er,			- 3
		004E . 0006	290 291	VSPTR PLEN	EQU EQU	>6E 02	San Barrier	**************************************		
		0008	292: 293	BLEN .	EQU .	PLEN+2		182		.*
			294 295		&RSEC	(UNIT, SECT	or,BUFFER,E	RROR)		
	5310 5320	635763	296	RSECT	DATA	#WSECT.#R	SECO1			
Ý	5321	054052 534543	297		DATA	5.:aRSEC:				
		066125	298	RSECO1	CALL	P				
		8F4C00	299		DST	>OOFF.aFA	C+2			
	5320					•				
	532E 5331	BF4A00 00	300	RSEC02	DST	>0000.2FA	С			
		B 04 003	301		ST	@Q1+1.@FA	C+2			
		BD4EQA	302		DST	aQ3.aFAC+				
		B05005	303		DST	@02.@FAC+				
•	533E	864000 0110	304		DST	>0110.RAM	(0)			
	5340	8F5500	305	RSECO3	DST	>0000.2>5	6			

, ...

1

di.

0

C.

1

>0111, RAM(0)

SET BIT "ON"

RSEC03

DATA #VON.#ONO1

DATA 3.: @ON:

DST

10/14/83

14:07:17

'ESM'

GPL ASSEMBLER 79.150

63A7 BFA000

63AF 63B763

6383 03404F

63AA 0111 63AC 056340 341

342

343

344

345

ON

CLOG RAM(801+2),801

1.8.9651

18714783

14:07:17

PAGE

Q

a particular com a transferencia com esta esta actual a proprieda de la proprieda de la compacta de la compact

6437 OF UE 6439 D802B0

6430 04

384

GUL ASSEMBLER 79, 450

5430	5443	335		es	UNSET	
643F	BFOEBF	385		DST	>BFFF, &Q4	
6442	FF					
	350008	387	UNSET	MOVE	8 FROM 204 TO RAM(203+2)	
6446	BOOCOE					
5449	050012	388		CALL	RPL	
	647C64	389	GETWRD	DATA	#PUTWRD.#GETWO1	
544F				ar and a	P 000 7 7 64 -	
	054047	390		DATA	5.:@GETW:	
	455457	201	CETTION	0.51.1	P	
	066125		GETW01	CALL DCLR		
	870A CBUADO	392	÷		ag3 .pt. >4	1
	046479	<b>4274</b>		# 9411 A San Sec	Charles There is a C	
	E000E0	394		OST	RAM(801+2).80	1
5454		ψ. γ. s				
	0465A1	395		CALL	CIF	1
		396			8 FROM 8FAC TO RAM(802+2)	1
	£0084A					
	9504	397		DIRCT	801+2	1
	A30800	398		DADD	8.802+2	1
6473						
6474	91UA	399		DINC	<b>@Q3</b>	1
6476	05645B	400		\$END		
5479	060012	401		CALL		
547C	64AB54	402	PUTURD	DATA	#GETBYT.#PUTWO1	
547F	చిత					
	054050	1 (37				
		403		VAIA	S.: &PUTN:	
6483	555457					
6483 6486	555457 066125	404	PUTWO1	CALL	P	
6483 6486 6489	555457 066125 870A	404 405	PUTW01	CALL DCLR	P aqs	. 1
6483 6486 6489 6488	555457 066125 870A CBOAOO	404		CALL DCLR	P	1
6483 6486 6489 6488 6488	555457 066125 870A CBOAOO 0464A8	404 405 406		CALL DCLR \$WHILE	P aqs aqs .DL. >4	
6483 6486 6489 6488 6488	555457 066125 870A 080A00 0464A8 350008	404 405		CALL DCLR \$WHILE	P aqs	1
6483 6486 6489 6488 6491 6494	555457 066125 870A CBOAUO 0464A8 350008 4ABOU4	404 405 406 407		CALL DCLR \$WHILE MOVE	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC	1
6483 6486 6489 6488 6488 6491 6494	\$55457 066125 870A 080A00 0464A8 350008 4A8004 0F12	404 405 406 407		CALL DCLR \$WHILE MOVE	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC CFI	
6483 6486 6489 6488 6491 6491 6497	\$55457 066125 870A 080A00 0464A8 350008 4A8004 0F12 8DB008	404 405 406 407		CALL DCLR \$WHILE MOVE	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC	1
6483 6486 6488 6491 6491 6497 6497	\$55457 066125 870A CB0A00 0464A8 350008 4AB004 0F12 BDB008	404 405 406 407 408 409		CALL DCLR \$WHILE MOVE	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC CFI aFAC.RAM(aQ2+2)	1 1 1 1
6483 6489 6488 6486 6491 6497 6497 6490	\$55457 066125 870A 080A00 0464A8 350008 4A8004 0F12 8D8008 4A	404 405 406 407		CALL DCLR \$WHILE MOVE XML DST	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC CFI aFAC.RAM(aQ2+2)	1 1 1
64836 64836 64839 64836 6491 6499 6499 6499	\$55457 066125 870A CB0A00 0464A8 350008 4AB004 0F12 BDB008	404 405 406 407 408 409		CALL DCLR \$WHILE MOVE XML DST	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC CFI aFAC.RAM(aQ2+2) aQ2+2	1 1 1 1
6483 6486 6489 6488 6491 6491 6497 6497 6497 6447	\$55457 066125 870A 080A00 0464A8 350008 4A8004 0F12 8D8008 4A 9508	404 405 406 407 408 409 410 411		CALL DCLR \$WHILE MOVE XML DST	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC CFI aFAC.RAM(aQ2+2) aQ2+2	1 1 1 1
6483 6486 6488 6488 6499 6499 6499 6499 6448 6448	\$55457 066125 870A CB0A00 0464A8 350008 4AB004 0F12 BDB008 4A 9508 A30400	404 405 406 407 408 409 410 411		CALL DCLR \$WHILE MOVE XML DST DINCT DADD	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC CFI aFAC.RAM(aQ2+2) aQ2+2 8.2Q1+2	1 1 1 1
64888814789006 644888814789006 6448999006 6449906 6444906 6444906 644496 644496 644496	\$55457 066125 870A CBOAOO 0464A8 358008 4AB004 0F12 BDBOO8 4A 97508 A30400 2 08 910A 6 05648B	404 405 406 407 408 409 410 411		CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC CFI aFAC.RAM(aQ2+2) aQ2+2 8.2Q1+2	1 1 1 1
648888814779006 6448888814779006 6444999006 64449844444444444444444444444444444444	\$55457 066125 870A CBOAOO 0464A8 358008 4ABOO4 0F12 BDBOO8 4A 9508 A30400 2 08 910A 6 05648B	404 405 406 407 408 409 410 411 412 413		CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL	P aQ3 aQ3 .DL. >4 8 FROM RAM(aQ1+2) TO &FAC CFI aFAC.RAM(aQ2+2) aQ2+2 8.2Q1+2 aQ3	1 1 1 1
83699881479006 644888897479006 64444444444444444444444444444444444	\$55.457 066125 870A CBOAOO 0464A8 358008 4A8004 0F12 BDB008 4A 9508 7508	404 405 406 407 408 409 410 411 412 413 414 415	GETBYT	CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL DATA	P aQ3 aQ3 .DL. >4  8 FROM RAM(@Q1+2) TO @FAC  CFI	1 1 1 1
83699881479006 644888897479006 64444444444444444444444444444444444	\$55457 066125 870A CBOAOO 0464A8 350008 4ABOO4 0F12 BDBOO8 4A 9508 4A 9508 4A 9508 4A 9508 6 05648B 6 05648B 6 060012 8 640064	404 405 406 407 408 409 410 411 412 413 414	GETBYT	CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL DATA	P	1 1 1 1
836988114790000 6448888114790000 6444899444990000000000000000000000000000	\$55.457 066125 870A CB0A00 0464A8 350008 4AB004 0F12 BDB008 4A 9508 7508	404 405 406 407 408 409 410 411 412 413 414 415	GETBYT	CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL DATA DATA	P aQ3 aQ3 .DL. >4 8 FROM RAM(@01+2) TO @FAC  CFI	1 1 1 1
8369881147900000000000000000000000000000000000	\$55.457 066125 870A CB0A00 0464A8 350008 4AB004 0F12 BDB008 4A 9508 7508	404 405 406 407 408 409 410 411 412 413 414 415 416	GETBYT GETB01	CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL DATA DATA CALL	P aQ3 aQ3 .DL. >4 8 FROM RAM(aQ1+2) TO aFAC  CFI aFAC.RAM(aQ2+2) aQ2+2 a.aQ1+2 aQ3  RPL #PUTBYT.#GETB01  5.:aGETB: P	1 1 1 1
836988114790062338888814444444444444444444444444444444	\$55457 066125 870A CB0A00 0464A8 350008 4AB004 0F12 BDB008 4A 9508 7508 7508 910A 605648B 605648	404 405 406 407 408 409 410 411 412 413 414 415 416	GETBYT GETB01	CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL DATA DATA CALL DCLR	P aQ3 aQ3 .DL. >4  8 FROM RAM(@Q1+2) TO @FAC  CFI aFAC.RAM(@Q2+2) aQ2+2 8.2Q1+2 aQ3  RPL #PUTBYT.#GETB01  5.:@GETB: P aQ3	1 1 1 1 1 1
836988114790063338888884444444444444444444444444444	S55457 066125 870A CB0A00 0464A8 350008 4AB004 OF12 BDB008 4A 9508 A30400 208 910A 6060012 6440064 85 7054047 455442 6064125 870A CB0A00	404 405 406 407 408 409 410 411 412 413 414 415 416	GETBYT GETB01	CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL DATA DATA CALL DCLR	P aQ3 aQ3 .DL. >4 8 FROM RAM(aQ1+2) TO aFAC  CFI aFAC.RAM(aQ2+2) aQ2+2 a.aQ1+2 aQ3  RPL #PUTBYT.#GETB01  5.:aGETB: P	1 1 1 1
8 6 9 8 8 1 1 4 7 9 0 0 6 2 2 2 8 8 8 8 8 8 8 1 1 4 7 9 0 0 6 2 2 2 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	S55457 066125 870A CB0A00 0464A8 350008 4AB004 OF12 BDB008 4A 9508 A30400 208 910A 6060012 6440064 85 7054047 455442 6064125 870A CB0A00 06640A	404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419	GETBYT GETB01	CALL DCLR \$WHILE MOVE XML DST DINCT DADD DINC \$END CALL DATA DATA CALL DCLR \$WHILE	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC  CFI @FAC.RAM(@Q2+2) aQ2+2 8.2Q1+2 aQ3  RPL #PUTBYT.#GETB01 5.:@GETB: P aQ3 aQ3 .DL. >8	1 1 1 1 1 1 1
3698 E1147900063388888888888888888888888888888889999AAAAAA	\$55,457 066125 870A CB0A00 0464A8 358008 4A8004 0F12 0F12 08 4A80400 2 08 7 910A 6 05648B 6 05648B 6 05648B 6 05648B 7 910A 6 05648B 7 910A 6 05648B 7 910A 7 910A 8 91	404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420	GETBYT GETB01	CALL DCLR \$WHILE MOVE  XML DST DINCT DADD CALL DATA CALL DATA CALL CALL CALL CALL CALL CALL CALL C	P	1 1 1 1 1 1
3698 E147900 F3258 BEF F1584 F44444444444444444444444444444444444	S55457 066125 870A CB0A00 0464A8 350008 4AB004 OF12 BDB008 4A 9508 A30400 208 910A 6060012 6440064 85 7054047 455442 6064125 870A CB0A00 06640A	404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419	GETBYT GETB01	CALL DCLR \$WHILE MOVE  XML DST DINCT DADD CALL DATA CALL DATA CALL CALL CALL CALL CALL CALL CALL C	P aQ3 aQ3 .DL. >4 8 FROM RAM(@Q1+2) TO @FAC  CFI @FAC.RAM(@Q2+2) aQ2+2 8.2Q1+2 aQ3  RPL #PUTBYT.#GETB01 5.:@GETB: P aQ3 aQ3 .DL. >8	1 1 1 1 1 1 1 1 1 1 1 1

'BSM'

GPL ASSEMBLER 79.150

£. :

6

10/14/83 14:07:17

PAGE 10

and the second s

GPL	. ASSEMBL	ER 79	1.150	/ Bs	ថ <b>ក</b> ។	10/14/83	14:07:17	PAGE	11
64C5	0665A1	422	-	CALL	CIF			4	. 1
	350008 80084A	423		MOVE		C TO RAM(a	Q2+2)		1
54CF		424		DINC	@Q1+2				ď
	A30800	425		DADD	8.802+2				1 1
6404	08								.8-
8405	910A	426		DINC	<b>ବ</b> ଢ଼ ଅ				1
6407	0554BA	427		\$EMD					^
540A	060012	428		CALL	RPL				
	551A64	429	PUTBYT	DATA	#RUM,#PUTB	01			
64EU									
	054050	430		DATA	5.:@PUTB:				
	555442								
	055125	431	PUTBC1	CALL	P				
54EA		432		DCLR	a				
	CB0A00	433		PMHILE	@03 .DL. >	8			1
	086509 350008	434		MOUE	W EDOM GAM	1001.03 TO	55.0		
	4ABOO4	404		MOVE	a FRON KAN	(801+2) TO	&F A C		1
	0F12	435		XML.	CFI				4
	808008	436		ST	aFAC+1.RAM	C000403			1
64FD				٦,	ornor thum	A CONTRACT ELECTRICAL			1
54FE	9108	437		DINC	802+2				1
6500	0040EA	438		DADD	8.801+2				1
5503	08								*
6504	910A	439		DINC	&Q3				1
5505	0564EC	440		\$END					
6509	060012	441		CALL	RPL				
	*	442	*						
	AE0608	443	BYTBIT	DIV	8.002			* * *	
	8608	444		CLR	802+2				
	800906	445	•	ST	aQ2,aQ2+3				
6514		446		CLR	aQ2	<u>.</u>	•		
6519	A10408	447		DADD	202+2,201+	2			
	uu	448 449	ж	RTN			A <sup>*</sup> ,		
	663065	450	RUN	DATA	#SAVE.#RUN	กา	. •		
6510		,		D	a	<b>U</b> 2	•		-
	035255	451	. '	DATA	3.:RUN:		•		
6521	4E			,					
6522	D7703F	452	RUN01	\$IF	aptop .DEQ	. >3FFF 60	TO RUNG2		*
	FF658A		,						
	800270	453		DST	aptop.ades	Τ .			
	9102	454		DINC	adest				
	B00002	455		DST	abest.asrc				
	8F083F	456		DST	>3FFF.aBLE	11			
6533	A50870	, ,,,,,,,,		F. (1) 2 ( 4)					
	BF703F	457		DSUB	aptop.agle	• •			
653A		458		DST	>3FFF.apTo	۲			
	A50008	459		DSUB	ablen.asrc				
	090030	46D			askc .DHE.				1
	4552	· 'wat yan'		77 77 1 8 de fee fre	* VIII	www.se.es			ì
	340880	461		MOVE	SELEN FROM	RAM(@SRC)	TO RAM(@DEST)		1
6546	028000						v a tem the feat total 3		•

on the

5

122°.

**(** 

8

W.

450

Tylo T

PC

÷.

\$25 \$25

GPL ASSEMBLER 79.150	'B9	m'	10/14/83	14:07:17	PAGE	12
	007	2000 2500T				1
6549 BD0200 462		asrc.adest				1
654C A50008 463	ดรบย	ablen.askc				i
654F 05653E 464	SEND					
6552 BD0A02 465	DST	adest.ads				
6555 A50A30 466		astln.aq3				
6558 B00230 467	OST	astln.ades				
6558 A10208 468		ablen,ades				
655E 340ABO 469	MOVE	ags from R	AM(@STLR) T	O RAM(@DEST)		
6561 028030						
6564 A13208 470	0000	SBLEN. SEKL	•			
6567 A13008 471		ablen.astl	1			
656A BUOA3O 472	,	EQG.NITEG				
6560 950A 473		<b>ଥ</b> ଉଅ		R TO LINE ADDR		
656F 010A32 474	#MHILE	203 .DLT.	@ENLN			1
6572 658A						
6574 350002 475	MOVE	2 FROM RAM	(803) TO 80	ļ		1
6577 00800A						,
657A A10008 476	DADD	ablen.ad				1
657D 350002 477	MOVE	2 FROM 80	TO RAM(8Q3)			1
6580 B00A00						
6583 A30A00 478	DADD	4.803				1
6586 D4						
6587 OS656F 479	\$END				, ,	
658A B03432 480 RUND2	DST	BENLN. BDAT	A SEI-UI	P FOR FIRST LIN	( E.	
6580 A73400 481	osus	ATAG6,ED<	•			
6590 03	7. C.L. D.	** * * * * * * * * * * * * * * * * * *	114570	ALL PARTITUE		
5591 87A3EC 482	OCLR			OW CONTINUE		
6594 BF6EOS 483	DST	VRANVS, avs	PTR SET ST	IACK ENPIT		
6597 F8	n. 01 m	AUGGTA NOT	renor			
6598 BD246E 4840	OST	avsptr.ast 1.abuffy	VSP1.		#	
6598 BE4401 485						
659E 052840 486		PRGLST	O CLOATING			
		T INTEGER T R IN 2>00	O PEOMITIO			
·		NG IN FAC,F	AC14 E7	A C 4.7 - 5 - 5 - 5		
489 * 65A1 490 CIF	EQU		PIGTAR PROPER	10 F 2 T	4.4	
		aFAC			. None	v.
65A1 874A 491 65A3 BD4COO 492		00.0FAC+2				
65A6 874E 493		aFAC+4	•	,		•
65A8 8750 , 494		aFAC+6				
65AA 8F4C 495	DCZ	aFAC+2				
65AC 65CE 496	BS					
65AE AF4A27 497		10000.aFAC	. *	1		
65B1 10	DOIA	1000010170	•			
6582 AE4C64 498	017	100.@FAC+2	<b>x</b>			
6585 BE4A42 499	ST	>42.aFAC	•			
6588 8E4B 500	CZ	&FAC+1		<i>(*</i>		
65BA 45CE 501	BR	CIFEND				
65BC 350003 502	MOVE		C+2 TO @FA	C+1		
658F 484C	FEDER T Nov.	ar triucts with				
65C1 A64A01 503	sue	1.@FAC				
65C4 8E4B 504	CZ	afac+1				
65 C 6 45 CE 505	8 R	CIFEND				
65C8 BD484C 506	DST	aFAC+2.aFA	AC+1			
65CB A64A01 507	SUB	1.0FAC				

10/14/83

14:07:17

PAGE 13

'BSM'

GPL ASSEMBLER 79.150

	•	550 551	**** CALL	_ @SAV	**************************************	ADDR) ****
663A	066125	552	SAVE01			
5530 5540	06 06	553		ST	6.RAM(ROLOUT)	
5541 5544	BDA3C6	554	SAVE02	DST	@01,RAM(ROLOUT+6)	LENGTH
	BDA3C2	555		DST	@Q2.RAM(ROLOUT+2)	ADDRESS .
	8F5603	556		DST	ROLOUT+9.2PABLOC	DSR PTR
5540	340AA3 CABOQE	557		MOVE	203 FROM RAM(204) TO RAM	(ROLOUT+10) FILE NA
		558	* 03 MUS	ST BE	LESS THAN 18	
6553 6556	8CA3C9 08	559		ST	8Q3+1.RAM(ROLOUT+9)	MAME LENGTH
5557	060010	550		CALL	CALOSR	CALL DSR
665A	80	551		DATA		
	050012			CALL		RETURN TO BASIC
	867266				#EXIT, #LOADO1	
6561	රසි					
	05404C	564		DATA	5.: aLOAD:	
	4F4144				_	
			LOADO1			
SAAF	8EA3CO 05	565				CODE FOR 'LOAD'
			*			
			EXIT	DATA	#SASUB.#EXITO1	•
	7C		***		,	<b>#</b>
		559	· 11 A	DATA	5,:@EXIT:	
55/9	584954		EXITO1	*** 10 7 7*		
00/0	UB	570	EXTIGI	E X I I		
			PROG1	DATA	#PROG2 PIR TO NEXT	
		572			#PROG11 ENTRY 20.:CPU PROG AT C(>6000)	
	144350 552050	3/3	en de la companya de La companya de la co	DATA	28.10PU PROG AS COMMON	
	524F47				•	
668A	204154				•	•
668D	204328				•	
6690	3E3630					
	303029					
6696	0F70	574	PR0611	XML	>70 CPU >6010	
6698	6688	575	PROG2	DATA	#PROG3 PTR TO NEXT	
669A	6681	576		DATA	#PROG3 PTR TO NEXT #PROG21 ENTRY	
		577			20.:CPU PROG AT G(>7000)	1 2
669F	552050					•
66A2	524847					
	204154					
55A8	204728					
55AB	3E3730					
56AE	303029					
65B1	311000	578	PR0621	MOVE	>1000 FROM ROM(#>7000) 1	TO @>9000 E2000-830
5684	8F9000					
65B7	7000					

GP1	L ASSEMB	LEK 79	.150	۶ ،	\$11	10/14/83	14:07:17	PAGE	15
5589	0F20	579 580	*	XML	>20	CPU >2000	)		
	0000 7000	581 582	PROG3	DATA DATA	#0 #>7000				
668F	144750 402050	583		DATA		PROG AT GC	>7000):		
66CS	524F47 204154					,			
65CB	204728 3E3730								
	303029	584	v	5" +10		•			
	•	204	*	EKO					

£.,

**(**)

(E)

50.0

(E)

**6**5

8

14:07:17

nnoo

0002

0004

0000

0001

0003

0008

Spell out ohrase FLaG sceech DATA ADdr Pointer To Last Char In List Time LEnght of timing char PHrom ADDRess

TEMPorary opu location 1
TEMPorary opu location 2

READ byte interface
Contains addr of speech periferal
WRITE byte interface
CONDition flag used in testing
PHrom DATA
Pointer To Current Byte Ext Data
Pointer to Last Byte Ext Data
Pointer To End Byte Ext Data
STRing LENght
BYTE 1
BYTE 2

Contains addr of speech periferal

BYȚE 3 SFeaK StaTuS

ABGG 9000 PODE 0010 0012 0014 0015 004A 0048 0040 0040 004F 61Ù PTLCIL EQU >4F 0051 TIMLEN EQU 051 611 PHADDR EQU >52 0052 612 613 0054 TEMP1 EQU >54 514 0056 615 TEMP2 EQU 056 616 0058 FOIL >58 617 READ 618 005A WRITE EQU >SA 619 620 005C 621 - COND EQU >5C PHDATA EQU >5D 0050 622 PTCBED EQU >5E 005E 623 0040 PILEED E00 >60 624 0062 525 PTEBED EQU >62 0044 626 STRLEN EQU >64 MAAA 627 BYTE1 EQU >66 0067 628 BYTE2 EQU >67 0048 EQU >68 629 BYTE3 0069 630 SPKSTS EQU >69

600

·-

-134

\*

Sometimes of the second second

ા જાત કહે તેને જૂના અજના જર્ફકાન હતી.

at a shi kalenda ka wakika afa

कार्यक्रीत १ ५००) । व्यवस्थानुं क्षेत्रीय क्षेत्रीय क्षेत्रीय

6

656

ि

6748 51 574C 874D

698

DCLR @DATAAD

'BSM'

14:07:17

10/14/83

GPL ASSEMBLER 79.150 'BSM' 10/14/83 14:07:17 PAGE 19
674E 066968 699 CALL STOATA
700 \$ERD
6751 00 701 RTN

<u> C</u>

'Bsm'

		703	PHRASE	
A752	D64A23	704		1
	478E		The state of the s	-
	9108	705	DINC SPTCCIS You bet we do!	1
	800860	706		2
	204764			
	9108	707	DINC APTCCIS	2
	056759	708	\$END	1
	8008	709	#1F RAN(OPTCC18) .EQ. NUMBER THEN Null phrase (all spa	-
	23476D	,,	چې ښاه د د د د د د د د د د د د د د د د د د د	
	9108	710	DINC &PTCCIS Yes, Go on to next char	2
676C		711	RTN lonore this onrase, oo get next o	
,	res was	712	#END	•
6760	BDOCUS	713	DST aptocis.aptFcip	1
		714	\$REPEAT	-
5770	9103	715		2
	BC4AB0	716	ST RAM(OPTCCIS).OCCHAR	2
6775				
	C5080A	717	\$1F &PTCC1S .OH. &PTLCIS GOTO FNDRUM	2
6779	6780			
	064A23	718	. SUNTIL &CCHAR .EQ. NUMSER	1
	4770			_
-,		719	FNDRUM	
6780	BD1008	720	DST APTCCIS.APTLCIP	1
6/83	9310	721	DDEC OPTLCIP	1
6785	9108	722	DINC APTCCIS Point to char after #	1
6787	066780	723	CALL GETTIN	1
	864C	724		1
678C	47AC	725	CLR &SPLFLG Clear soell-flag (don't soell) #SELSE	1
678E	800008		DST aptccis.aptfcip	1
	CA4A41		\$IF OCCHAR .HE. ASCIIA THEN 1st char alfabetic?	2
	479A			
6796	864C .	728	CLR @SPLFLG Yes, clear spell-flag (don't spell)	2
	4790			2
679A	BE4001	730	ST >1.83PLFLG NO. SET SPELL-FLAG (DO SPELL)	2
		731	#END	
		732	\$REPEAT	
6790		733	DINC aprocis	2
679F	0667BD	734	CALL GETTIN	2
57A2	D651FF	735	SUNTIL ATIMLEM .ME. OFF PTCC WILL POINT TO FOLLOWING (	. 6
67A5	679D			
	801008	736	OST aprocis.apricip	1
67AA	9310 .		DDEC apticip	1
			\$END	
67AC	00	739	RTM	

GPL	_ ASSEMBLE	R 79.150	* <b>285</b> 11 *	10/14/83	14:07:17	PAGE 21	
	4.	$\mathcal{A}_{i} = \{ x_i \mid x_j = 0 \}$					
	*				•		
	and the first of the second	741 TIMING					
67AD	CAS1FE	742 \$RHI	LE &TIMLEN , NOT.	.HE. >FE			1
6780	678C			,			
67B2	A14851	743 DA	DD STIMLEN. STOTT	T1M	·		1
67BS	9108	744 DI	MC aptccis	•		•	1
6787	066780	745 G. CA	LL GETTIM				1
678A	47AD	746 #SEN	LE STIMLEN .NOT.  DD STIMLEN.STOT  NC SPTCCIS  LL GETTIM  D	•	1 to 1		-
67BC	00	74 <u>7</u> RTN					
		748	- 				
	•	749 GETTIM	•				
67BD	BC4ABO	750 ST R	AM(APTCCIS).acci aptccis,atempi	HAR .			
67C0	0.8						
67C1	BD5408	751 DST	aptccis, atemp1	•			
67C4	C5540A	752 - #1F	atemp1 .DH. apri	CIS THEN			1
67C7	47CD						
67C9	BES1FE '	753 ST	>FE.@TIMLEN				1
67CC	00	754 RT	F4				1
		755 . \$€พบ					
57CD	C&4A3B	756 \$IF	acchar .notH	. SEMICO THEN			1
670C	682F		N &CCHAR .NOTH F &CCHAR .EQ. SI				
6702	D64A20	757 \$1	F acchar .Eq. S	PACE THEN			2
6705	470E						
6707.	BES104	758	ST >06.aTIMLEN CALL SKPSPC Sk:	•			2
670A	066840	759	CALL SKPSPC Sk:	to over all ex	tra spaces		2
0/00	UQ .	7 G U	KIN		•	•	2
1.5.2		761 . ∌E	NO FACCHAR . EQ. PI				
O/DE	D64AZB	762 \$1	F &CCHAR .EQ. PI	LUS THEN			2
6/61	4/EU .					*	
0/60	7154	/63	DING WIEMPI Ne	ed to test the	next char		2
6/15	U66833	764	CALL NUMERC IS	it numeric		7	2
675A	୍ <b>ପ୍ରୟଟ</b> ୍ଟେଟ୍ଡୋଗ - <b>୧୯୯୪</b> -	705	DINC OTEMP1 New CALL NUMERC IS BS NOTIME Was	numeric, there	iore not a ti	ming char".	2
. 07EM	0031	(00 7(7)	CLK GITHER NO	, set as notin	ırna	*	2
_G/EG		/O/( <u>13,33</u> 5,335,335,335,335,335,335,335,335,3	NO	وأنتسا والمرافق والمستحراط المراج	The second s	المريبة بالمستوي	Z
67ED	066420	760 FE	E SCCUAR EA	SMMA TUEN 3	ana	egi i kalagot sagat kan	market to do the
67ED	47F4 % 6 1 %		FRANCISM FEWER STATES	ZORIM IDEN I SE ZOROZNIKA KANTA			
- A7F2	BF5112	77n 800000	ST 342 STIMIEN		LOUIS AND AND SALES		A Commence
47F5	ກດ	771	RTM	Suppried Transport of the Control of		A STATE OF STATE OF	<b>4</b> (4 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
		777	MD	and the second s			4
67F6	D64A2E	773 \$I	F ACCHAR .FQ. P	FRIOD THEN	the state of the s		?
67F9	480D					The same of the same	\$ says to the contract of the
67FB	9354	774	DDEC STEMP1 Go	back to prece	ding char		2
67F0					CARING CHEST		2
4800	066833	775	CALL NUMERC IS	TE DIMERTOR			
ದಾಬಿಟಟಿ	.066833 6 6 6 4809	775	CALL NUMERC IS	it numeric?	and the second of the second		2
6802	.066833	775 776 777	BS NOTIME Was CLR OTIMEN NO RTN NO F SCCHAR .EQ. CALL NUMERC IS BR PTIME DINCT OTHER NOTICE CALL NUMERC IS BR PTIME	it numerice es. on to foll	owing char		2.
680£	066833	///: 778	CALL NUMERC IS	es, on to loll it numeric to	owing char ·		2
680£	066833	///: 778	CALL NUMERC IS	es, on to loll it numeric to	owing char ·		2
6804 6807	7554 066833 682F	778 779	CALL NUMERC IS BS NOTIME Both	it numeric to it numeric to were numeric.	owing char o? therefore no	ot a timing	2 2 8
6802 6804 6807 6809	066833 682F 8E513C	778 779	CALL NUMERC IS BS NOTIME Both ST >3C.OTIMEN	it numeric to it numeric to were numeric.	owing char oo? therefore no numeric, the	ot a timing erefore a ti	2 2 & 10
6802 6804 6807 6809	066833 682F 8E513C	777 787 780 780 PTIME 781	CALL NUMERC IS BS NOTIME Both ST >3C.OTIMLEN RTM	it numeric to it numeric to were numeric.	owing char oo? therefore no numeric, the	ot a timing	2 2 & 10
6804 6807 6809 6800	066833 682F BE513C	777 778 779 780 PTIME 781 \$	DINCT BIENPT YOUR CALL NUMERC IS BS NOTIME Both ST >3C.BTIMLEN RTM ND	es, on to foli it numeric to were numeric, Both were not	owing char oo? therefore no numeric, the	ot a timing erefore a ti	2 2 2 2 2
6804 6807 6809 6800	066833 682F BE513C	777 778 779 780 PTIME 781 \$	CALL NUMERC IS BS NOTIME Both ST >3C.OTIMLEN RTM	es, on to foli it numeric to were numeric, Both were not	owing char oo? therefore no numeric, the	ot a timing erefore a ti	2 2 & 10
6802 6804 6807 6800 6800 6810	7554 066833 682F 8E513C 00 064A2D 481D	777 778 779 780 PTIME 781 \$	DINCT BIENPT YOUNGER IS BS NOTIME BOTH ST >3C.BTIMLEN RTM ND F &CCHAR .EO. H	es, on to lold it numeric to were numeric, Both were not	owing char oo? therefore no numeric, the	ot a timing erefore a ti	2 2 2 2 2
6802 6804 6807 6809 6800 6800 6810 6812	7554 066833 682F 8E513C 00 064A2D 481D 9154	777 778 780 PTINE 781 \$E 782 \$E 783 \$I	DINCT BIENPT YOUR CALL NUMERC IS BS NOTIME Both ST >3C.BTIMLEN RTM ND	es, on to lold it numeric to were numeric, Both were not  YPEN THEN eck next char	owing char oo? therefore no numerio, the	ot a timing erefore a ti	2 2 2 2 2
6802 6804 6807 6800 6800 6810 6812 6814	7554 066833 682F 8E513C 00 064A2D 481D 9154 066833	777 778 779 780 PTINE 781 782 \$E 783 \$1 784 785	DINCT BIENPT YOUR CALL NUMBERC IS BS NOTIME BOTH ST >3C.BTINLEN RTN ND F &CCHAR .EQ. H DINC &TEMP1 Ch	es, on to lold it numeric to were numeric, Both were not  YPEN THEN  eck next char it numeric?	owing char oo? therefore no numerio, the	ot a timing erefore a ti	2 2 2 2 2 2 2

1

(E)

60

(6)

mi.

5.	GPL ASSEMB	LER 79,150	'BSM'	10/14/83	14:07:17	PAGE	22
	( o e o o o o o o o o						
F 45	6819 BE510C	787	ST >OC. aTIMLEN	was not numer:	ic. therefore	set as	tim2
	681C DD	788 .	RTN				2
<b>6</b> 78		789	#ENO				
<u></u>	6810 D64A3A	790	\$IF &CCHAR .EQ.	COLON THEN			2
	6820 4826						
-61%	6822 BE5130	791	ST >30.aTIMLEN				2
	6825 OO	792	RTN				2
		793	<b>WEND</b>				••••
	6826 064A3B	794	\$IF &CCHAR .EQ. !	SEMICO THEN			2
40	6829 482F						-
	682B BE511E	795	ST >1E.&TIMLEN	•			2
41.4	682E DO	795	RTN				2
		797	\$END				-
,		798 4	END				
			INE				
	682F BE51FF		T >FF. aTIMLEN Se	t as no timino	char present		
	6832 00		TN		mort. w. m. pr. m. w. a./ h. 17 %		

**Q** 

G,

\$IF &BYTE1 .DNE. O GOTO TRYAGN

10/14/83

14:07:17

'BSM'

GPL ASSEMBLER 79.150

O.S

€.

587D 8F6548

6880 58 6831 8740

6883 00

839

840

842

Laurence of the Constitute of

841 FOUND

MTR

DCLR @DATAAD

68F8 01 68F9 4900

$\sim$			e , , , , , ,	27 A D & U
. ·	/ 8 5 /	0000000	844 5	PST &PTF&PH,&PHADDR LGADAD must have addr in PHADDR
		805212	845	CALL LOADAD Set up the Phrom to scan a word
Š.		0,6691E BEC000	846 847	
Marie 18		5A10	047	31 /10,00 (WRITE)
			848	CLR &PTLCPH Save the length of the word in 2 b
<b>5</b>		8616 861700	849	ST 80(READ).8PTLCPH+1 at PTLCPH
Ź.,			047	21 OF (VERO) OF LEGICIES AND LINES
		0058	850	DADD SPTFSPH, SPTLCPH Then add front pointer giving end
<b>6</b>		A11612		DST APTEBPH. APTCCPH Set up current byte pointer into Phro
S. S. L.		BD1412	851 852	DINC OPTCOPH 1 byte beyond the length byte is first char
		9114	853	DST OPTFCIP.OPTCCIP Reset current byte pointer into phras
<b>.</b>	5075	BDOEOC		Try comparing chars
100				EXT
	4004	BECOOO		ST >10.20(WRITE) Read a byte from Phrom
Ø.		5A10	030.	GI FIGURETEN REED & STORY FILLS
<b>V</b> G/*		BC50C0	857	ST &O(READ),&PHDATA Get data in from the Phrom
		0058	U J r	or delicably arrows and account are arrows and are
Ç.		045080	858	\$1F &PHDATA .EQ. RAM(&PTCCIP) THEN Compare Phrom with Pha
*1.		064890		<b>***</b>
		9114	859	DINC APTCOPH Equal, advance both pointers 1
6		910E	860	DINC APTCCIP 1
•		DABOGE	861	\$1F RAM(@PTCCIP) .EQ. SPACE THEN Need to skip extra spa
		204807		
		06684D	862	CALL SKPSPC 2
		D6E001	853	#1F RAN(1(PTCCIP)) .EQ. NUMBER THEN 3
	68C1	0E2348		
<b>E</b>	68C4	C 7		
	68C5	910E	864	DINC apticip
	'		865	\$END
Ø.	237		866	SEND FEND
	58C7	C51416	867	\$IF &PTCCPH .DH. &PTLCPH THEN Scanned all char in Phrom
i.		48F0		
		C50E10	868	\$IF &PTCCIP .OH. &PTLCIP THEN Yes, scanned all char 3
Ŀ		48E8		# - BB -
et n		805216	869	DST aPTLCPH, aPHADDR Yes, Word found 3
0		A35200	870	DADD >6.aPHADDR Move 5 bytes down from last char to
i.	6807		***	
(4)		066950	871	CALL READAD This will set up data addr in DATAAD 3
		BECOOO	872	ST >10.80(WRITE)
_		5A10		ST AD(READ) ASTRIEN
(i)		BC64C0	873	ST @O(READ),@STRLEN 3
260x		0058		RTN 3
	68ES		874	RTN 3 \$SELSE No. Phrase is longer, follow the higher link 3
	•	48EE	875	· · · · · · · · · · · · · · · · · · ·
Mar.		075200	876	DS1 NO. DELUMBER 11046 2 PACES DEAGING LIESLIN CO GEO CO
	68EB	; 4900	87 <b>7</b>	BR NXTPHR 3
学的	005(	ព្រះប្រាក	878	\$END
~c.6"	LAFE	48FB	879	\$SELSE 2
		. 45,5 ) C50E10	880	\$1F OPTICIP .NOTDH. OPTICIP GOTO NEXT Go compare &
<b>4</b> 0%		7 656210 5 48A1	- www	control of the second of the s
~		BF5200	881	DST >1.aPHADDR Move 1 byte beyond PTLCPH to get to 2
				· · · · · · · · · · · · · · · · · · ·

		-99-		
		883	\$E11D	
ARER	491E		\$SELSE	
	C45DBO	885		PTCCIP) THEN Phrom word is before2
	DE4909	44.2	** ** ** ** ** ** ** ** ** ** ** ** **	and a graph of the standard of the standard that the standard of the standard
		683	DST 31 SPHANDS Go	1 byte beyond last char to lower 12
6906		000	por Frankov da	I byte beyond rost that to lower in
		887	\$SELSE Phrase word	comes before Phrom word 2
	BF5200	888	DST >3. PHADDR Go	3 bytes beyond last char to higher?
690C	03			
		889	\$END	
		890	NXTPHR	
6900	A15216	891	DADD &PTLCPH.&PHADDR	1
5910	066950	892	CALL READAD Now get	the new addr 1
6913	8F4D49	893		THEN Any more leaves on this tree2
6916		** - **		
6917		894	RTN	No. return empty handed 2
567 / A 6	50			
		895		Yes, set up to check next word
5918	801240	896	UST @DATAAD,@PTFBPH	Store new addr in PTF8PH 1
691B	056884	897	B SEARCH Go compare	this new word!

'BSM'

GPL ASSEMBLER 79.150

10/14/83 14:07:17

PAGE 25

10/14/83

14:07:17

PAGE 26

'BSM'

GPL ASSEMBLER 79,150

- N.

600

8

G	PL ASSE	MBLER 7	9.150	'esm'	10/14/83	14:07:17	PAGE	27
		972	SPEAK					
λοα	7 114608			SETRU	•			
	A C9000	ህ ንውን ጉ ወደለግ	GREE GUILI	LE OPTFBSL .NOT	ALC APTIBE	•		1
	D 6A13	2 740	# 441.1 T	LE OFIFDSE INVI	vnc. oritoa	L		7
	F 066A1	F 941	C.A.	LL WAIT				1
	2 06880		C A	F RAN(OPTFBSL)	SIED SECTION OF LIFE KI			
	5 FF69B		#1	r KMH(GrirbSL)	.AE. FF INCA			2
	3 FF070 8 BC79B			ST RAM(@PTFBSL)	STIMES			2
	6 60776 8 00	U 743		of Mantarirese)	* CLTLIEV			£.
	°E 00 °C 8279	944		NEG STIMER				2
	E BDS2E			DST RAM(1(PTFBS				2
	1 0100	U 743		DSI KANCICETEDS	L//, OFTHOUR			4.
	3 A3000	0 946		DADD 3.@PTFBSL				2
	15 ASUUU 16 03	0 740		DANK S.GRIFESE				ú
	10 03 17 02790	n 977	1.00.04	\$IF ATIMER .LT.	0 6010 10004			2
	A 49A7	W 771	E001 I	er arribal acre	0 0010 LOO11			<del>****</del>
	C 8E526	9 948		\$IF @PHAUDR .NE	о тығы			3
	F 88	, , , , ,		+ x 1				u
	 30 06891	F 949		CALL LOADAD				3
	3 BECOO			ST >50.80(WRI	TEY			3
	6 5A50				, , , , , , , , , , , , , , , , , , , ,			-
	·	951		\$END				
69E	88 4A10	952						2
	A 9100				kis >FF			
	C BOSEB	0 954		DINC APTEBSL S DST RANGAPTEBSL	). DPTCBED Set	. uo pointer t	to front	byt2
698	F'00					, , , , , , ,		·
690	0 06805	E 955		\$IF RAMCAPTOBED	)) .NE. >60 GOT	O ERR1 Inval:	id speech	stż
	3 60488						1	
				CLR &PTEBED		4 4 4 4 4 4		2
690	08 BC63E	F 957		ST RAM (-1 (PTCBE	D)),aPTEBED+1	Get total le	enght of	strl
690	B FFFFS	E '			1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	And the second		
69(	CE A1625	£ . 958		DADD @PTCBED.@F				
690	1 9500	959		DINCT OPTFBSL	Skip addr byte	15		2
		960	DIRSPH	₹		•		
691	)3 BEC00			ST >60,80(WRITE	) Start Speak	External		2
	06 5A60		the same and	DINC APTOBED S	and the same of the	And the second second	7.5	
	08 915E		2 to make	DINC &PTCBED S	ikip speak exte	ernal command	्रीतिक क्षेत्री है	2
691	OA BD608	0 963		DST RAM (OPTCBE	),aPTLBED Set	up pointer t	to last b	yte2
	DD 5E							
	DE 955E			DINCT @PTCBED	Skio lenght by	tes, this is	first re	eal fi
691	EO A1605	E 965		DADD SPTCBED. SP	TLBED Add ler	ant to first	byte giv	/ing2
				ST 16.8TEMP2				2
	E6 D55E6	2 967	LOOPR	\$IF ∂PTCBED .	DEQ. APTEBED 6	OTO CONTIN		2
	E9 6A10							
	EB DSSE4	0 968		\$IF @PTCBED .	DEQ. @PTLBED T	THEN		3
	EE 49F6							
	TO 066A1			CALL WAIT		* · · · · · · · · · · · · · · · · · · ·	•	3
691	-3 05690			B DIRSPH		•		3
		971		\$END				
	-6 BCCOC			ST RAM(@PTCB	ED), au(WRITE)			2
	79 5ABOS							
	FC 915E	973		DINC SPTCEED				2
	FE 9256			DEC STEMP2				2
sΑ.	DO 49E6	975		BR LOOPR				2

**∌REPEAT** 

GPL AS	SEMBLER 79	.150 '59M'	18/14/83	14:07:17	PAGE	28
6A02 066	A14 977	CALL GETSTS				***
6A05 DA6			Senvere en a			3 2
		SUNTIL .BIT6 &	15rk515 .E.G. 1			4
6A08 6A0		OT 6 ATEMPS				_
6ADA BES		ST 8.STEMP2		a.		2
6A00 056		B LOOPR				2
		ΦEND		•		
	982	CONTIN				
6A10 056		<b>事</b> E pd D				
6A13 00	5'84	RTN				
	<b>925</b>					
	986					
	987	GETSTS				
6A14 BC6	900 988	ST @0(READ),@SPKST	rs			
6A17 005	8					
6A19 BC6	900 989	ST @O(READ),@SPKST	rs			
6A1C 005	8	•				
6A1E 00	990	RTK				
	991					
	992	WAIT				
	993	\$REPEAT				
6A1F 066		CALL GETSTS				1
6A22 DA6		·	era do n			
6A25 4A1		worktam whats dulite	ara amy a ta			
4827 BB		BTN				

Œ

(F)

100

6

Sec.

€

\$1F &CHAT .EQ. COMMA\$ GOTO DIRSPK Now another word stri

and the second s

**'888'** 

10/14/83

14:07:17

PASE 29

6PL ASSEMBLER 79.150

6A79 D642B3 1045

The second secon

GPL ASSEMBLER 79.150

'BSM'

10/14/83

14:07:17

PAGE 30

647C 644F

O

1046 \$END

5A7E 056987 1047 CALL SPEAK Go say all that is on the Speak list

6A81 OF18 6A83 4B5C 1048 XML VPOP Get rid of the string space

BR RETURN Return to main program 1049

GPL ASSEMBLER 79.150 'BSM' 10/14/83 14:07:17 1051 1052 1053 Get parameter and check for string arguments 1054 1055 GETPAR 1056 6A85 8E8089 1057 \$1F &GROMFL .EQ. O THEN Norking on VDP program 6A88 4A8D 6A8A B02056 1058 DST @FAC+12.@CPU2C and wodate the program pointer 1059 ⊈END IF XKL PGMCHR 6A80 OF18 Get next token 1060 6A8F D642B7 SIF ACHATINE, LPARS GOTO ERR2 Quit if no left parenthesis 1061 5A92 4BC3 6494 00 1062 RIN 1063 1064 GETPRM 6A95 OF18 XML PGMCHR Get next token 1065 PARSE RPARS 6A97 0EB6 1065 \$IF &FAC+2 .NE. >65 GOTO ERR3 Not a string, something is 6A99 064C65 1067 SAPC 4BCS 5A9E 00

Return normal

```
1070
             1071
                          SPGET subprogram. Load speech data from external
             1072
                          device. Use standard file I/O.
             1073
             1074
             1075
             1076
             1077
                   SPGET
6A9F 066A85
                     CALL GETPAR
            1078
                                            Skip header and "("
             1079
                   MXTPAR
            1080
6AA2 066A95
                     CALL GETPRM Get parameter
6AAS 8F506B 1081
                     #IF OFAC+6 .DNE. #0 THEN found non-mull ASCII string
6AA8 57
                       DST &FAC+4.&PTFCIS Set up pointer to first oher in stri
DST &FAC+6.&PTLCIS Set up pointer-to-last-char-in-strib
6AA9 80054E
            1082
6AAC B00A50 1083
6AAF A10A06 1084
                       DADD OPTFC15.OPTLC18 by adding lenght-of-string to pa
6AB2 930A
            1085
                       DDEC SPILCIS
                                                first-char-in-string and subtrat
5AB4 066980 1085
                       CALL SETRA
6A87 BDC806
           1087
                       DST AFTFCIS. APTCCIS
6ABA 066780
           1888
                       CALL GETTIM
SABD 8548
             1089
                       CLR STOTTIM
6ABF 0667A0
            1090
                       CALL TIMING
6AC2 CS08GA 1091
                       $1F SPICCIS .NOT. . DH. SPILCIS THEN
6AC5 6B57
5AC7 056752
            1092
                         CALL PHRASE
5ACA D64CG1
            1093
                         #IF OSPLELG .EQ. 1 THEN
SACD 4AD2
5ACF B0100C
                           DST &PTFCIP, &PTLCIP
            1094
             1095
                         SEND
6A02 066859
            1096
                        CALL LOOKUP
6AD5 8F4D4A
            1097
                         $IF ODATAAD .DEQ. O THEN
6AD8 ED
6A09 BF4D71
            1098
                           DST >71F4.2DATAAD
6ADC F4
6ADD 8E6451
            1099
                          ST >51,@STRLEN
                                                                                3
             1100
                         #END
            1101
SAEO BD0140
                         DST &DATAAD. &PHRADD
                                                                                2
6AE3 BC0064 1102
                         ST OSTRLEN. OPHLEN
6AE6 A20003 1103
                         ADD >3.8PHLEN
                                                                                 2
6AE9 D642B3 1104
                         $1F 8CHAT .NE. COMMAS GOTO ERR2
SAEC 48C3
SAEE OF18
            1105
                         XML PGMCHR
6AFO OF13
            1106
                         XML SYM
6AF2 0F14
             1107
                         XML SHE
            1108
6AF4 OF17
                         XML VPUSH
6AF6 8600
            1109
                         CLR add
6AF8 BC0D00 - 1110
                         ST APHLEN. SOD
                                                                                 2
6AF8 060038 1111
                         CALL GETSYR
                                                                                 2
6AFE 066980 1112
                         CALL SETRW
6801 BF4A00 1113
                         DST SREF. aFAC
6804 1C
5805 BE4C55 1114
                         ST >65.0FAC+2
6808 BD4E1C 1115
                         DST OSREF. OFAC+4
680B 8650
             1116
                         CLR aFAC+6
6000 8C5100 1117
                         ST OPHLEN. aFAC+7
```

GPL	ASSEMBL	ER 79.	150	'BSM'	10	1/14/83	14:07:17	:	PAGE	33
	BEEDOO	1118	9	or >60.Rancoc	SREF))		·			2
6813 6815 6818	85E001	1119	(	LR RAM(1(SRE	(F))					2
	BCE002	1120	9	ST APHLEN, RAM	(2(SREF)	)				2
681E	A3E001 1CFFFD	1121		)ADD -3.RAM(1	(SREF))	-				2
6524	E05201	1122 1123		ST &PHRADD,&	PHADDR					2 2 3
	C60003	1124		WHILE APHLEN	.н. з	,		•		3
682F 6832	BEC000 5A10	1125		ST >10.20(W	(RITE)				•	3
6834 6837	BC58CO OO58	1126		ST @O(READ)	.aBYTE3					3
6839 6838	8567 BE5408	1127 1128		CLR @BYTE2 ST >8.0TEMP	1					3 3
683E 4841		1119	RMDAG	DSRC OBYT						3
6845				DSLL @BY1	ΓΕ1.#1					3
6848		1132		DEC &TEMP1 BR RNDAG						3 3
684D	BCE003			ST aBYTE1.		EF))				
684F 6851	9200	1134 1135	•	DINC @SREF DEC OPHLEN						3 3 2 2
6B53	482A 0F15	1135		≢SEND XML ASSGNV ND				2.	. <u>.</u>	2
4007		1139	\$END		3555A E	TO SIVEDA	.,, D			
685A				achat .eq. co	JIMP GU	IV HAŞTA		and the second		
 685C	D64286 -	1142	. \$IF	OCHAT .NE. RI	PARS GOT	O ERR2			Sec.	•
6861	OF18 060012	1143	XML	PGMCHR RPL						

**(1)** 

CAT

**(** 

```
GPL ASSEMBLER 79.150
                                 18SM1
                                             10/14/83
                                                         14:07:17
                                                                       PAGE 34
                           004.0.0.0084.0084.0.0 32AS
              1146
 6866 15A9AC 1147 ERR$1 DATA 21,:ILLEGAL SPEECH FORMAT:
 6869 ACA5A7
6860 A1AC80
 686F 8380A5
 6872 A5A3A8
6875 8OA6AF
6578 B2ADA1
 6878 84
 6870 13A9AE
             1148 ERR$2 DATA 19.:INCORRECT STATEMENT:
 687F A3AFB2
 6B82 B2A5A3
 6885 848083
 6888 84A184
 688B ASADAS
 688E AEB4
 6890 168384
             1149 ERR#3 DATA 22.:STRING/NUMBER KISMATCH:
 6893 82A9AE
 6896 A78FAE
 6899 BSADA2
 6890 A58280
 659F ADA983
 65A2 A0A184
 6845 A3A8
 68A7 16B3B0 1150 ERR$6 DATA 22,:SPEECH STRING TOO LONG:
 6BAA ASASA3
 SBAD ASSOBS
 6880 B482A9
 6883 AEA780
 6BB6 B4AFAF
 6889 8DACAF
 ABBC AEA7
              1151
                           BASE 0,0,>300,>300,0,0,0
              1152 FRR1
 688E 06001C 1153
                      CALL ERR$$
. 68C1 6866
                      DATA #ERR$1
              1154
              1155
              1156 ERR2
 68C3 06001C 1157
                      CALL ERR$$
 68C6 687C
              1158
                      DATA #ERR$2
              1159
              1160 ERR3
 6BC8 06001C 1161
                      CALL ERR$$
 68CB 6890
                      DATA #ERR#3
              1162
              1163
              1164
                    ERR6
 6BCD 06001C 1165
                      CALL ERR$$
 6BDO 68A7
              1166
                      DATA #ERR$6
 5802 000065 1167 ENDLNK DATA #0.#EXIT
 6805 72
              1168
                              END
  ERRORS=
  LENGTH= 3030 (>0506)
```

260 SYMBOLS USED

LINK TO DSR ROUTINE.

>DA PAB LENGTH, STANDARD

0010 51 DSRENK EQU >10.

52 \*
53 \* EQUATES FOR PERIPHERAL ACCESS BOCK

سنج بمرابر الأنجيات الأصيحة وسلكانا حياه والمنتجاء كا

```
GPE ASSEMBLER 79.150
                                                                    10/17/83
       $0000 $55 SCREEN EQU >00
                                                           SCREEN OFFSET FOR PAB
       003C 57 MAXLEN EQU 60 MAXIMUM LENGTH OF FILE NAME
1000 58 PAB1 EQU >1000 PAB ADDRESS USED FOR ALL FILE
1080 59 BUF1 EQU >1080 BUFFER FOR PAB1
                                                         PAB ADDRESS USED FOR ALL FILE TRANSIT
BUFFER FOR PAB1
      1000 50 PAB1 EQU >1080 BUFFER FOR PAB1
1100 60 PAB2 EQU >1100 Second PAB
1180 61 BUF2 EQU >1180 Second BUFFER
1200 62 PAB3 EQU >1200 Third PAB
    1280 63 BUF3 EQU >1280 Third BUFFER
1300 64 PAB4 EQU >1300 Fourth PAB for
1380 65 BUF4 EQU >1380 Buffer also us
65 * CPU RAM EQUATES
                                                                Fourth PAB for COPY statement
                                                                 Buffer also used to load progrm file
                     67 * CPU RAM EQUATES
        0000 69 TEMP EQU >00
                                                                Temporary storage area
         0002 70 TEMP2 EQU >02
0003 71 TEMP3 EQU >03
0004 72 TEMP4 EQU >04
         0002
                          73 . *...
         73 *
0010 74 0SR EQU >10
0018 75 CHRBUF EQU >18
         0010
                                                              6 bytes temp loc for general load ro
Char buffer for blinking cursor
                                                 >10
76 *
77 SREF
                                                          Pointing to the PAB start address. flag and status byte
                                        - EQU
                                                >1 C
         4.0
101 *
0049 102 FLAG2 EQU >49
0000 103 LD EQU 0 Bit 0 on = loader is being executed
0001 104 VAR EQU 1 Bit 1 on = save with var 80 format
                        101 - *
  | 104 | VAR | EQU | 1 | Bit 1 on = Save with var 80 format | 105 | *EDT | EQU | 2 | Bit 2 on = Error in editor | 106 | OTHR | EQU | 3 | Bit 3 on = Flag for "other" program | 107 | *ETOP | EQU | 4 | BIT 4 ON = EDITOR MEMORY WRITTEN OV | 108 | GPL | EQU | 5 | Bit 5 on = GPL program is called from | 6006 | 109 | LDUTIL EQU | 6 | Bit 6 on = first load during LOAD & | 110 | *
```

```
Gf:0H
     GPL ASSEMBLER 79.150
                                  10/17/83 09:15:56
  004A 111 FAC EQU >4A

004C 112 FAC2 EQU FAC+2

004E 113 FAC4 EQU FAC+4

0050 114 FAC6 EQU FAC+6

0057 115 FAC10 EQU FAC+10

0056 116 FAC12 EQU FAC+12

005C 117 ARG EQU >5C Used in conversion of int to float
      113 * STANDARD EQUATES FOR GPL INTERPRETER
   120 *
0070 121 MEMLEN EQU >70
0073 122 SUBSTK ECU >73 Subroutine stack pointer.
0074 123 KEYBRD EQU >74 Keyboard number to scan.
   007A 131 MPC EQU >7A
007C 132 SYATUS EQU >7C
133 *
134 ******
```

```
PASE .
     GPL ASSEMBLER 79.150
      155 ****
   156 * Initilazation section

157 * Color >F5 (color = 1)
            157 * Color )F5 (color table 0 through 12), blue screen 158 * Define VDP registers, load character sets,
   158 * Define VDF registers, load character sets,
159 * No potion and no sprites
140 ****
161 *
   162 BEGIN2
6029 066525 163 CALL COLOR Load color table
602C BE7405 164 ST 5.2KEYBRD Scan kayboard 5.
602F 03 165 SCAN Enable keyboard for /4A
             165 SCAN ENBOYE REYBURD AND 166 SIF OKEYBRD EQ. O THEN MUST BE 4A KEYBOARD

167 DST >BOU.OFAC
   6030 8E7440
    6033 30
   6034 BF4A08 167
   6037 00
6037 00
6038 06004A 168
6038 4056 169
                      CALL CHAR3$
                                       Load lower case character
                      CALL CHAR35
  603D BEAADA 170 0
                      DST >ACO.8FAC
  6041 060018 171 CALL CHAR2$
                                        load semi-lower character
   6044 BF4A0B 172 DST >800.0FAC
                                       First character definition
  6047, 00
 6048 BEB04A
6048 CO
              173 CHRLP ST > CO.RAM(&FAC) Place two dots for lower case 1
  604B C0 174 DADD 8.2FAC
               1.
175 $IF &FAC .DL. >COO GOTO CHRLP Fix next character 1
 604F 08
6050 CB4A0C
6053 004048
184 *
6069 390007 185 : MOVE 7 FROM ROM(#VDPRG2) TO VDP(1) SET UP VDP REGIST:
606C 016A4B : CHCEK IF ERAM EXISTS
 ***** ERROR IN IMMEDIATE FIELD LINE= 187 COLUMN= 17
  ***** ERROR IN BRANCH ADDRESS LINE= 187
*606F 000000 187 ** CALL BUG1 TEST IF ERAM EXISTS. **
188 *
188 *
 6072 310010 189 MOVE 16 FROM ROM(#CDATA) TO CHAR(>1E) LOAD CURSOR CHA
   6078 38
              193 *******************
              194 START
```

```
GPL ASSEMBLER 79.150
                                            10/17/87
                                                       09:15:56
                              196
           197 *
                          Initial screen to select editor, assembler, loader
  176
198 ****
6079 BE737E 199 ST >7E.@SUBSTK Adjust subrouting stack pointer
607C 8722 200 DCLR @ERRCOD Clear error code
607E 8748 201 DCLR @FLAG
   5080 0720 203
                       ALL SPACE
                       FMT XPT=2, YPT=1, ':* EDITOR/ASSEMBLER * : ':.
                204
                                20,114,':PRESS:::1,10,274;
                205
                             (:1 TO EDIT: (.12,234;
:2 ASSEMBLE: (.12,194;
:3 LOAD AND RUN: ,10,154;
                205
                204
 208
                209 ':4 RUN!',24(,13)
210 ':5 RUN PROGRAM FILE:',67,11(;
211 '>A',':1981 TEXAS INSTRUMENTS:'
   4082 08FF02 -
 6116 03
                213 SCAN10 SCAN 214 BR SCAN10 215 $IF BKEY .EQ. BACK THEN
   6119 D6750F
   5110 411F ...
   611E 08
                            EXIT ....
               215
217
                        FEND
 217 $END
611F A67531 218 SUB >31,2KEY Convert from ASCII
6122 CA7505 219 $IF BKEY .HE. 5 GOTO SCAN10 Check input range
6125 6116 CASE BKEY
 6127 4133 221 BR EOLT Load a file
6128 4632 222 BR ASSEM Save a file
LAST ERROR FOUND AT LINE $187
 ***** ERROR IN IMMEDIATE FIELD LINE= - 224 COLUMN= 16
   ***** ERROR: IN BRANCH ADDRESS LINE= 1224 1. 12-2. 224 1. 224 1. 224 1. 224 2. 224 2. 224 2. 224 2. 224 2. 224
6131 4557 225. BR OTHER A Run other program
```

```
10/17/83
                                                                                                                                                       09:15:56
                                                                                          äROM
                   GPL ASSEMBLER 79.150
                2278*
   6136 8748 235 DCLR &FLAG
236 FMT XPT=2,YPT=2,':* EDITOR *:',10,22<;
      247 *
6198 03 248 SCAN70 SCAN
619C 4198 249 BR SCAN70
619C 06750F 250 $IF OREY .EQ
                                                                          SETE OREY .EQ. BACK GOTO START
     61A1 6079 3 2 2 3 3 SUB >31.8KEY
261 * * CODE FOR PURGE INSTRUCTION ( ) * *
                    262 ******************************
              6188 D67504 263 $IF akey .eo. 4 THEN 618E 41E4
  61BE 41E4
61CC 08A112 264% FMT 20. ARE YOU SURE(Y/N)? ( 1 1 4 1 5 2 4 5 6 1 C 6 20594F)
          61C9 $52053
61CC $55245
61CF 28592F
        61CF 28592F
61D2 4E293F
61D5 20FB
61D7 05679F 265 CALL YORN
61DA D67559 265 $1F aKEY .EQ. :Y: THEN
61DD 41E2
61DF 066549 267 CALL CLRTOP Clear top of
                                                                                                                                             Clear too of memory for edita
             A CONTROL OF THE CONT
                                                                                                                                          61E2 4133 269 BR EDIT
270 SEND IF
```

PARE

GPL ASSEMBL	ER 79.150	GROM	10/17/83	09:15:56	PAGE 7	*
						April 1
	Transact Salaran	*********				Salah Salah
		_ Take care of SA				
		**********				.1
61E4 D57503		\$IF @KEY .EQ. 3	the state of the s	· ·	The state of the s	
61E7 61FC						Yr.
	275 *	###### (1 : 1 : ### 1 ) 1 :				
	and the second of the second o	oad editor				
	277 *	Check if editor	is already t	here(temporar	O	
61E9 D78F9D	278	. \$1F albcobe .be	Q. >55AA GOTO	OPEN		
61EC 0055AA				•		
61EF 61FC						*
	279 *	LOAD EDITOR		4 W 1		3 4
61F1 0665B2	- 280	CALL DOWNLD	CALL DOWN L	OADER	and the state of the state of	
61F4 6623	***	DATA #EDNAME		1	9	
61F6,0665D4			LOAD, IT, DOW			
51F9 066549	283	CALL CLRTOP	CLEAR	EDITOR INFORM	MOITAN	4
and the second second	284 **	the second second	* ***			
* 4	285 *					
CACO DA COLONA	285 OPEN					
	287	\$1F &TEMP4 .EQ.	1 GOIO EEDII	GO TO EDITOR		
61FF 6265	്രമാ ച	CHECK FOR VARIA	 	m or calle	•	
6201 060402		#1F aremp4 .eq.		SAVE FILE		
6204 4237	1407 1407	arr dichra .cv.	Zinca	SHAF LIFE		
		FMT XPT=2.YPT	=18 '-VAP 80	FORMATIV/N12	· /	
6209 FE1213		7 711 7 71 7 7 7 7 7 7 7 7 7 7 7 7 7 7	-101 11HK 00.	Vidira Ciray:	•	
620C 564152	The state of the s	The state of the s	The second secon	The second of the second s		
620F 203830						
6212 20464F	<b>建一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个</b>					
6215 524041						
6218 542859						
621B 2F4E29	and the second seco	and the second of the commence of the second	and the manufacture of the month of the state of the stat	and the second s	and the control of the second	and the department of the description of the second
621E_3F20FB	as to confirm the hemically is a first because a factor confirmation.	grafia (1994). Karangan mengangan kanangan mengangan pendangan pendangan pendangan pendangan pendangan pendangan pendangan pe	ويوافظ فالماري والموارية والموارية والمارة والمارة والمتارة والمارة والمارة والمارة والمارة	manana in manana parahasa in manana and basis and and a	and a supplication of the	Carlos regions of the contract
6221 06679F	Contract the contract of the second substitution of the	CALL YORN	- GET YES OR	NO ANSWER	The second second second second	The second
6224 D67559	292 Shinn	* * 1F, aKEY/.EQ.	Y: THEN		2.000	iose e Cris
6227.422C	A SCHOOL SECTION	AND SB & SFLAG2.				
M. 0117 DO+7U2 @			VAK: NOW SEIN VA	KIABLE% FORMA I	rLAU ( William Zo Lyn	Sandarate, Service
622C D6750F	294	*END IF * SER SKEY .EQ.	DACK COTO CC	TT CA DAGE T	C COLT COURT	
622F 6133	473	Pirant I Cv.	BACK GOID EL	ITI AO DACU	O COTI POKER	a interpression of the Kr.
6231 BF2002	796	DST >282.aLC	CIUTA ANTUS	THE CHRECK		1.0
6234 82						
プムウスログ ムウスログ	3707 2 3 30 30 5	SELSE OF THE	監備 经股份公司 医多性糖	ost victorii Reili iliaa miidd	6 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
6237 BF2002	298	DST >222.ALC	SPECIE	Y CURSOR LOCA	TION 1	
623A 22						
	299	\$END IF				
6238 31000A	300	MOVE 10 FROM RO	M(#FLN) TO RA	m (aLoc)		
623E B02062		DST >222.8LC		1.77年代中国的发展		
6241 5B	rtigether kind o			等。[2] [2] [2] [2] [2] [2] [2] [2] [2] [2]		
5242 A32000	301	DADD >20.8LOC.	CURSOF	R FOR FILE NAM	E.	
6245 20		· · · · · · · · · · · · · · · · · · ·	Account to the second s	マーキ アーティー (報2 1 年) ほど	the state of the s	to a service of the s
6246 BF1C10	302	DST PAB1.85REF	en e		and the second of the second o	, i.,
6249 00		DST PAB1.@SREF CALL DSR1 PAB is all cles				
624A 0663A0	303	CALL DSR1	Get th	ne file name a	nd PAB	
	2 <b>304</b> € <b>*</b> ∾.⊘	PAB is all cles	ared. so it is	ın open mode		

GPL ASSEMBLER 79.150 GROM 10/17/83 09:15:56 PAGE 8
6240 060402 305 \$IF ATEMP4 .EQ. 2 GOTO ESAVE
6250 6285
6252 060403 306 \$IF ATEMP4 .EQ. 3 GOTO PRINT
6255 6209
6257 8E0462 307 \$IF ATEMP4 .EQ. 0 GOTO ELOAD
625A 72
308 \*\*\*
625B 46494C 310 FLN DATA :FILE NAME?:
312 \*\*\*\*\*

```
GPL ASSEMBLER 79.150 GROM 10/17/83 09:15:56 PAGE 9
  315 *************************
  626C 016271
626F 4133 322 BR EDIT
6271 E0 323 ON DATA >E0
 6271 EO
          324 3.*
        325
          326 * ELCAD ---- routine to load a file
          327 ****
  328 ELOAD
6272 066278 329
                 CALL FIXVAR . Try to open in fix and var forant
  330
331
             **********
           332
             * LOADER FOR SOURCE FILES
           333 ***
           334
             ***
  335 * Note: End of file will be checked when I/O error happens
336 * while reading a record passed end-of-file.
337 * See ERRIO section to see how load gets finished.
627E 1CO4
6290 DAE001
629A 02
629B 00 361 RTN No error, go back to the caller
629C BCOOEO 362 TRY ST RAM(FLG(SREF)). STEMP Copy flag byte
629F 011C
```

```
GPL ASSEMBLER 79.150
                      GROM
                              10/17/83
                         62A1 B2001F 363 AND >1F. DTEMPS Delete the error code
 62A4 060004 364
               $1F BTEMP .EQ. 4 THEN
 62A7 42B3
62A9 BEE001 365 ST &00010100, RAMIFLUID.
62AC 1C14
62AE B64902 366 SB @FLAG2, VAR Variable 80 format now. 1
  62A7 42B3
                 SEND IF

BR ERRIO Otherwise error!!!
           370
          371
                 ****************
           372 *
                   SAVE SOURCE FILE
           374 ESAVE
 6285 BEE001 375 ST 02.RAM(FLG(SREF)) Output mode
 6288 1C02
 628A DA4902 376 $IF .BIT(VAR) @FLAG2 .EQ. 1 THEN
6280 6204
628F 8EE001 377
                 ST >12.RAM(FLG(SREF)) Var 80 output mode 1
 62C2 1C12
           378
379
                   $END IF
 62C4 0663E9
                  CALL DSR3
                                        Open it.
          380 *
          381 ST 03.RAM(@SREF) Store OP code for write
 6207 BEBOIC
 62CA, 03
 62CB BEEOOS
          382 ST 80, RAM(CHRCNT(SREF)) Store character count.
                              62CE 1C50
     303. *
384 * Write one record at a time.
385 *
386 WRITE
 6200 OF22 387 XML SAVEX Execute SAVE in editor.
6202 6707 388 BS EMSG SAVE error
6204 0663E5 389 CALL CLOSE Close the file
6207 4133 370 BR EDIT Go back to editor screen
391 *
 6202, 67C7
 6204 0663E5
6207 4133
 840/6
           ั้ 393 🖫
              * PRINT ROUTINE TO PRINT OUT FILES
          399 *
             * Inout file is defined. Now get output device
           401
620C O8FF02 403 FMT XPT=2, YPT=21, : DEVICE NAME?:
 62EF 8F2002 405 0ST >2c2,aLoc 405
62F3 BF1C11 406
               DST PAB2, @SREF Open next file
 62F6 00
                  CALL DSR1
```

GPL	ASSEMBL	ER 79	.150		GROM	10/17/	83	09:15:56	PAGE 11	
				3 (A 2 )				1446 - 2011		
6302	545043							and the state of the	ાર્ફ્કુલ કોલ્ડ કેલ્પ્ટ્રેસિયા કુરોનો સ	No. 15
5305	OA S									
	86E004	410	and the second	CLR	RAM(LEN(SRE	EF)) Ch	ange 1	ogical reco	ird length 1	
6309	1C					a da bara	ring. Na hiji kalendari			
6 704	DEEDO4				IF				80 dsiplay	and the second
7	BEE001	1 <b>41 2</b> 8							au deiblay	
	BFE002	413	28 6 7 1 6 7 1 6 1 6 1 6 1						ne buffer are	
10.0	101080									
the transfer of	0663E9	414			DSR3		Open i			
1 44 A	D6E004	415		\$IF F	RAM(LEN(SREF)	EQ.	>20. TH	IEN OIG TP!!	11 / P   P   14	
	102043	A Table				William I				4
	22 864802	416		6 ( S) ( S) ( S)	aFLAG.TP	elektronik	Set a	flag	4	
0011	en-der	417	er in the	\$END		4.	0e4 a	1160	*	
6322	BEBO1C	418	Maria de la como de la		03.RAM(@SREF	)	Ready	for print!		
6325	03 - 8	1. 1. 1. 2.1.	Modellin	3500	The Market Bay of the	经发生数				. *
15 16		419	*	Section 1						
	DA4802	420	N.	\$IF	BIT(TP) &FLA	3 .EQ.	O GOTO	FAST RS2	232 case.	
6329	6380									
4.7700	954640	421	PRINT1	DST	0404 00000		**=		. wi	
	8F1C10	422	Control of the	051	PAB1. &SREF		иом де	t first fil	<b>(皇</b>	
A .	044300	dia a i		CALL	CLRBUF	,	Clean	up buffer a	areas - All Carlos	1 3
A Committee of the Comm	the state of the s	43.4			DSR3				and the second s	1.00
6335	BF1C11	425	garante e e e e e e e e e e e e e e e e e e	DST	PAB2, @SREF		Precar	e for print	the state of the s	Mr. 1507.
6338	over the same of the same	and the second	and the second seco	elektrika (m. 1914) 1885 - January Marian, 1914	pro-more a proming and make a recording	and the particular and	د المام المحمد المام المام وقد المام الم	and the same of th	and the second s	
A CONTRACTOR		426	electric and registration	No.	and the second second					8.5
	All the second	427 428	THE PERSON LIE CONCEPTION ASSESSMENT AND	Next	routine chec	ksythe. Kodobakin	bulfer	and replac	ce characters	ali etalogi Berapakan
A339	BF0010	429		DST	BUF1, aTEMP 🛠	ី 2 CH	eck hi	iffer area	rational territories	86 N
Sec. 145	80		2. 2. 2. 2. 4. 4. 4. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.			Arministra A.			A CONTRACTOR OF THE CONTRACTOR	18-200
6330	A30000	430		DADD	79. aTEMP	Ch	eck fr	on end of i	record	ditta   Kahamai
6340										
6341	.BE0250\$	431	A STATE OF THE STA	ST	80.aTEMP23	ြို့က် <sub>ကို</sub> S∈	t the	counter	the state of the s	
Forkering A			LOOP	all Mark	er installet devia ja	Barana y		· 1000年3年2		
	BC0380*			<b>5</b> (5) (2)	RAM(aTEMP),a	IENTS	et cn	aracter code		1.00
and the same of the same of	- D60320	434		SIF	S.OS. ENNSTE	PACE GO	TO MO	ena de la composición dela composición de la composición dela composición de la composición de la composición de la comp		ero ne r <del>adio</del> nama.
Religious for the sections where	6361	The second	e de la companya de l	Kaliford C.						and the second
6340	060300%	<b>ं 435</b> ⋈	467784	\$IF (	атемрз %.ео.∞>	OC THEN	I II	oage feed s	kio lines Ja 1	17.15
6350	435F/		www.	被被称为	OF BUSINESS OF BUSINESS		Tall Supplies of the Said	XX 使的多数以其	material actions of	i it
was reason to the same	BEBOOO	∜.436		® ST	SPACE, RAM (	aTEMP)	Put	space in. 🖔		
6355	20						alian da marinda da 1980 da marinda da mari			
	044350	437	e balla termina Managaria Managaria	CAL	L DSR3	$rac{1}{2}$ $rac{1}{2}$ $rac{1}{2}$	ıst sk	io a few li	nes: 1	
* 6357	0603E7	2 4 3 G	A STANCE	CA	I DONO	9 B		ag Tabungkaya	y gradi kalinggana katalang a sa at	harry -
To the Desire		440		SEND.	IF	等的证据			int out.	4 4 4
∵ 635F	4367	441	医多种基本	BR S	PRINT2	No r	need to	o check. or	int out Addition	
	24.85 F. L. C. W.	. 6.62	24 (1) 12 12 11	Annual Control of the	and the second of the second o					1
6361	9300	443		DDEC	atemp	Decreme	ent, the	e buffer po	inter	
6363	9202	444	Owner Le	DEC	atemp2	Decrema	ent co	unter	And the second of the second o	
6365	4544	445	***	BK 🤼	aTEMP aTEMP2 LOOP PAB2.aSREF	Not bas	ssed ti	ne last loc		
		447	*	⊼ ⊼ ⊼ ₹ ; ; . DST :	PAR2 ASPEE	Δ	ready	specified		
1 M 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	and the second of the second o	• • • •			The second of the state of the second of the	A				100

0

6

€.

0 ...

0000

18 (18) 48 (18)

	GPL	ASSEMBL	ER 79.	.150	GROM	10/	17/83	09:15:56	PAGE	12	1
		v akon na	//8	PRINT2	and the second s						
	4347	0663E9			L DSR3		Call to	o write one	lime	an Talange,	
, · · · · .	0001		450	the result and the state of the			00.12 0	J 772 JUL 0110	12110		
**		magnetic and property that the same			llowing so	ection ta	akes car	e of Thermal	Printer	case	
	o and a referal Total Supple		452,4	**************************************		TO THE THEORY		的方式的基础的基础	类数数数数数	ar 32 \$ 8	inar v
	636A	A60220	453	SUB	⇒20,aTEM	P2	F.	ind how many ore bytes to	more to	move	
(see the			¥454	\$ \$ \$ \$ \$ IF.	atemp2 .	GT. O THE	ЕМ 💢 🥻 М	ore bytes to	eprint?	1	
		4380	ان سرسور	The second of the							
, f		A3E002	455	, D	ADU 220.R	ė́u (Rnt (2)	KEF)) SI	hift buffer	adoress	. 1	
٠		100020	456	ing. Ting dia kabupatan 😄	т : Этемр	O Ramiteli	OCHT/CPE	F)) Put cour	<b>,</b>	entra 🛊 🗀	
		1002	120		1		NON CONE				
		056367		В	PRINT	<b>2</b>	P	rint one mor	e line	1	Section 1997
·			458	\$EN	D IF		F 25				
	6380	BFE002	459	DST	BUF1.RAM	(BUF (SRE	F)) R	eadjust poir	nter		
	6383	101080				Land to the state of the					
		BEE005		ST	>20,RAM(	CHRCNT (S	REF)) R	eadjust char	racter cou	nt	4
		1020	and the same		00000				1 V 1		
	6365	4328	461		PRINTI		Print	more line		`	
		•	463	***							Č
				FAST					**		
4	638D	BF1C10	465		PAB1.25	REF	Read	a record ( /	You know the second	<i>(</i>	
Service Services			환하철보								
1 2 40	6391	066300	466		L CLRBUF	3-4-6	Cean	up buffer		Back File.	
1.1	6394	0663E9	467	CAL	L DSR3			and the same of the same	Agricultural de la Companya del Companya del Companya de la Compan	Salah Sa Salah Salah Sa	and the second of the second o
- 1.5		BF1C11	468	DST	PAB2,85	REF	, Frint	_it_out	ki Bahari meri e mengantan	gang pangangan kanalang mengang bang	
		00			DEED L	Part of the second	Strain Control		Stable Second Comment		f i spilite versione. Seriesking en en gel
	27 4. 20 4. 14	0663E9		CAL		Could to the state of	3. 强、发 · 数 · 数 · 数 · 数 · 3. · · · · · · · · · · · · · · · · ·			4/2010/00/00	Paramatan an
	Later to the second for	.4380	. 470 »		* FAST			tathe proces		B. Carlotter	
*********	Comment of the second	Agenta and the same		* ° OSR		Programme Company of the Company	一种 曹二郎 网络人名阿尔马斯克				M. C. S.
				*****					*****	***	And the second
		and the second		DSR1		a Projekt	NE LIVER	Party 1, 5 (2) (2) (4)	* 5 * 5 * 1 * 4 * 1	Říkaní.	
	63A0	066300	475	CAL	L® CLRBUF	多名をさい	Clean	up the PAB	bûffer are	a 8 🛞 🔅	de la companya de la
				OSR2 ARE					a-9 Skorat a .		of Kathalian
- 7 N			C 389 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	CAL	L CLRPAB		Clean	up PAB area			<b>建设于64条</b> 等
23.6	and the same of	BDEO02	478	C DST	asref.R	AM(BUF(S	REF))	Find buffer	address		
1 Av. 2		1 C1 C	479	nΔn	NAG DAC	(BILE CEDE	FII D	uffer addre			
1 1980	M	100080	Combine was about			SARE SARE		STATE TOUR		4.50.224	was a second
" The second	As a market	BEEOO8	480	A to the second and the second	SCREEN.	RAM (SCR (	SREF))	Screen off	set ////	A Compellation	
		1 COO+2					The second in the world have be		· 数 15年 年 、 第二条 " 不 " " Y 、 Y 、 A (A) (A) 、 (A)	N. D. Jan.	
	6386	BFE004	481	fall DST	>5000,R	AM (LEN (S	REF))	Record len	gth 🌲		gelget in de de in de de. Gegenerationes
	6389	1c5000		and the second conservation of	A Time to the work to the control of	a to an arrival man in		Get the na	Same of the first of the second of the secon	The state of the s	4.2 Marie III. de la composición
	63BC	06640A	482	CAL	L NAME	The Commercial of the	San San San	Get the na	me/	. Comment	gradient de la companya de la compa
in industrial	6386	8F5063()	483	\$1E. 3	FACA DNE	. O THEN	, Do	not move 0	bytes. C	1	
	0362		484	COBVIN						in (10 mari). Desta de 10	
	6303	3450F0	485	ONE THE CORP.	E REACA F	ROM RAM	REACA) T	O RAM(PARIF	N(SRFF)) N	eggeriete Lament	Salarah da Kada
1.0	6306	OA1CBO	2001 <b>7 7</b> 1				waa 11 wa∓ / 1	O RAM (PABLE		- mwa.	
	5.409	A. F					4.4	A 100 March 100			
	63CA	BCE009	, 486	ST.	@FAC6+1	. RAM (NLE	N(SREF))	Put name 1	ength in P	AB 1	Martin de la Companya
	63CD	1C51						Put name I		480 GR (	
	The second		. 487	\$ SEND	IFm (Still)	Mark Market	₩ <b>4</b> j Jen W		21 多数的数据15		
		المراج فالبيسة	24	والمهري العراق والأمالة السا	La company of the contract of		Section 1	The second secon	-1	*	•

GROM

10/17/83

09:15:56

PAGE

12

GPL ASSEMBLER 79.150

0

•

6

6

0

(1)

```
63CF 00
                   489 ****
                 490 ****
491 CLRBUF
                                DADD >80.2SREF
   6300 A31C00 492
                                                           3 SREF= buffer address
    6303 80.3
                          ST SPACE.RAM(@SREF) Clean up the PAB buffer.
     6304 BEB01C 493
     6307 20
                   494 MOVE 79 FROM RAM(@SREF) TO RAM(1(SREF))
     6308_35004F
   630B E0011C
630E B01C
                               DSUB >80.aSREF
                                                            asref= PAB address
     63E0 A71000
                    495
  63E3 80
    63E4 00
                    497
                    498
                         *************
                 499
                               CLOSE A FILE
                    500 ******************
                    501
                    502 CLOSE
    63E5 BEB01C 503 ST_ 1,RAM(@SREF)
                                                            Store close on code.
     63E8 01
  504 DSR3
63E9 BD561C 50S
63EC A35600 506
                              DST askef.afac12
DADD PABLEN-1.afac12
Pointer to name length.
     63EF 09
 63F0 060010 507 CALL DSRLNK Call DSR routine
63F3 08 508 DATA 8
63F4 6871 509 BS ERRIO File not found
63F6 DAE001 510 CLOG >EO.RAM(FLG(SREF)) Any other error?
63F7 1CE0
520 *****

521 **** Get file name from screen

522 * FAC4 = points to file name in VBP

523 * FAC6 = 2 bytes name length

524 *

525 NAME

640A 06643D 526 CALL SCAN50 Get user input. file name

640D BD2024 527 DST ASTLN. DLOC Relocate to scan file name.

6410 BE003C 528 ST MAXLEN, DTEMP Set the counter.

6413 8750 529 DCLR OFAC6 Clear length of file name.

530 SKIPSP

6415 DARROD 531 SIF RAM(QLOC) .EQ. SPACE THEN Skip spaces
    6415 D6B020
6418 20442B
                    $15 RAM(QLOC) .EQ. SPACE THEN Skip spaces
```

5ROM

GPL ASSEMBLER 79.150

8

09:15:56

10/17/83

GPL	. ASSEMBLE	R 79.150	GROM	10/17/83	09:15:56	PAGE 14
			a to produce the second		•	
641B	9120	532	DINC aLOC	Check	mext location.	. 1
6410	9280	533	DEC ATEMP	Decrem	ent counter.	1
	4415		BR SKIPSP			
	B02024	**	DST `aSTLN.8LOC	Input	file name agair	1. 1
6424	DA4804	536	*1F .BIT(NOTHIN)	aflag .EQ	. 1 GOTO NAMENO	No input
	443C			1.0	exp <sup>20</sup>	·
		537	BR NAME	- No fil	e name on the	line 1
			\$END IF			
6428	B04E20	539	DST &LOC. &FAC4	Starti	ng address of :	file name.
		540 COUNT				
542E	068020	541	\$IF RAM(@LOC) .NE.	SPACE THE	N	1
6431	20643C			, ,		
6434	9150	542	DINC @FAC6	Increm	ent name length	1
6436	9120	543	DINC ALOC	Look f	urther.	1
6438	9200	544	DEC ATEMP	Decrem	ent counter.	1
643A	442E	5.45	er count	Chack	more in the li	ne. 1
		546	<b>≢END 1F</b>			
		547 NAMENI	)			
643C	00 -	548	RTN		•	

10/17/83

09:15:56

GPL ASSEMBLER 79.150

PAGE

		A STATE OF THE PROPERTY OF THE	The second of the second s	
GPL ASSEMBLER 79.150	GROM	10/17/83	09:15:56	PAGE 16
6497 4133	1944			
595 6499 D67509 596	\$END IF \$IF akey .eo. >0	O EATA PET	*** (	n d on bo de
649C 647C	PIF GRET .EV. VU	ופא טוטש ל	. *** Cursor I	right.
649E D67508 597	\$IF aKEY .EO. >0	8 THEN	*** Cusor la	oft. I was a second
- 15. 24.366 <b>6481</b> 3. <b>4486</b> 5.74.666.628.8888.8888		조건 ~ 6일 경치 경험 다시는 소설에 되었다면 다시다.	天我 花式大翻印刷 花鸡牛用鲱 医内脏失败 化二十二二	
64A3 8E0164 598	\$IF aTEMP+1 .E	Q. O GOTO SCAN	51 Can't go le	eft any mo
64A6 46 , 64A7 9320 599	DDEC aLOC	e i di ingan senjah bili ji se	Move cursor	to the 16
6489 4446 600	BR SCAN51		Go back to s	
1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	SEND IF			Mile County Things of the
64AB 067500 602	\$1F aKEY .EQ. >0	D GOTO SCHEND	*** Enter 1:	ine
5 64AE 6503	TO SEE SEEMS OF SEE			
6480 067503 603 6483 4403	\$1F @KEY .EQ. >0	S IMEN	*** Delete	key 1.
64B5 BEB020 604	ST SPACE.RAM	aLOC)	Store blank	1
6488 20 ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )		435		
6489 B00026 605	DST @ENDLN.@TE	MP	Find how man	ny bytes <b>t</b>
64BC A50020 606	DSUB aLOC, atem		en e	<b>1</b>
64BF CE0100 607 607 64C2 4446	\$IF aTENP+1 .L	E. U GOTO SCAN	151 Nothing to	the right
64C4 340080 608	MOVE STEMP FRO	M RAMCLECCO	TO RAM(RLOC) 1	Move resti
64C7 20E001		A TO THE	, , , , , , , , , , , , , , , , , , , ,	
64CA 20 00 4			rain and a second	
	ST SPACE, RAM	(QENDLN) / 多字页	Delete last	character
64CE 20	DDEC BENDLN		in a series de la companie de la co	
64CF 9326 610 64D1 4446 611	BR SCANS1		Adjust end (	
612				
6403 D67504 613	\$IF &KEY .EQ. >C	14" THEN A CAR	*** Insert	keyi
64D8 B64801 614	SB &FLAG, INSRT		Turn on fla	1
6408-4446 615 616 INSO			g GO DACK to	scan
64DD BD0026 617 8 3	DST @ENDLN.at	EMP.	Two lines f	illed? 11
** 64E0 A50024 6.618	DSUB astln, ate	MP & TOTAL STATE	Find length	of file bey
64E3 C6013C 619 .	\$IF aTEMP+1	I. MAXLEN GOTO	SCAN51 No spa	ce to inse
64E6 6446 73 620	DST BENDLN, BT	CHO	Find bytes	
64EB A50020 621	DSUB aLOC. aTEM	IP 22 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Find byces	
64EE 9100 622	DINC atemp		Number of b	ytes to mo
64F0 3400A3 623	MOVE STEMP FRO	M RAM (aLOC) TO	RAM (>3CO) M	ove to tem
64F3 COBO20 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3				
64F6 3400E0 624	MOVE STEMP FRO	M RAM(>3CO) TO	) RAM(1(LOC))	Move back1
64F9 0120A3 64FC C0 64FD 9126 625 64FF 4470 626	The control of the co	mage ( ) A mage ( ) mage ( ) man ( ) mage ( ) ma		
64FD 9126 625	DINC SENDLN		Ad just end	of line 1
64FF 4470 626	BR DSP		Activate th	e characta
627	SEND IF			
6501 4446 628	BR SCAN51	kanggar ing kanggaran bahasa	ignore othe	r control
6501 4446 628 629 SCNE 6503 00 630 631 ***	RTN	eg ve kumana sa kana ya sama sa Majara	Finish ente	ning mama
631 ***		TRIVER, CHARGE A		* * भ्रम् । १० शब्द ≠
	and the second s			
	and the same of th			
		And the second	•	,
	4	*		
and stated with the stated of	and a comparation of market	ili (191 <mark>4) de la partir d'i</mark> mparation de la company de la	and the same of	A Commercial States and the second of the second se

•

	, GPL ASSEMBLE	ER 79.150	GROM	10/17/83	09:15:56	PAGE 17	
		i de la companya del companya de la companya del companya de la co	and the second of the second o				* •
		635 *****		9	a fear way a con-		
7 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -		636 CHKRAM		the specific terms of the second			
. s. • ***	6504 BC008F	637	ST &RAMTOP.&	TEMP Save d	ata first		
4 1. 4	6507 9000 <u></u>			医乳腺性畸形 化电路电池性 医多种性贫疽	나를 하는 이번에 가는 하는 가 가는 사람들이 하는 것	and the contract of the contra	
	6509 BE8F9D	638	ST >FF, &RAMT	OP Try to	write to ERAM	to see it	
	650C OOFF		· · · · · · · · · · · · · · · · · · ·		· 		
**	650E D68F90	639	\$IF @RAMTOP #N	E. >FF GOTO ER	RRAM		
	6511 00FF48	No service of	Service Services	Salah Sa	Service Services	e e e e e e e e e e e e e e e e e e e	
	6515 868F9D	640	CLR GRAMTOP	Try ao.	ain with 00 to	make sure.	
*	6518 00					4	
	6519 8E8F9D		#IF @RAMTOP .N	E. OO GOTO ERRI	RAM		
	651C 004831						
	651F BC8F9D	642	and the second s	MTOP Save i	t back		
		643.	RTN		1.		
		644 *	14 1 14	l nad li	TILITY routine	s to FRAM.	
		645 ****		2000			
		646 COLOR					
	ii ii	647	ST >DO.RAM()	300) Stoo s	orites		
			/ : . <u></u>				
		648	ST >F5,RAM(>	380) LOAD C	OLOR TABLE		
	6520 35001F	649	MOVE 31 FROM R	AM(>380) TO RA	M(>381)		
71!	6530 A381A3						
1.00	4533 80°	· 表现的 在一个人的					
	6534, 0720	650	ALL SPACE				
	6536 04F5	651.	BACK >F5				시간 (1967) (31 전설
• 1 PASIG	and the state of t	653 ****	BIN PROCESSIONANA Procession				\$15 MM
		654 COLOR2	and the second s			dan sagaran salam bangan dalam di dada di Salam banga banga bang sa sa sa salam sa s Salam sa	and the same of
Togeton marchine	6539 BEA380	655	ST >13, RAM(>3			m Cara Cara	age of the second
				\$200 \$200 \$200 \$200 \$200 \$200 \$200 \$200	The second of the second of the second	which the state of the same is	esaco -
्रक्ति । विकास विकास		- 656 c - 1	MOVE 31 FROM R	AM(>380) -TO RA	M(>381)為大學基準		326
	6540 A381A3			Accessor to the second			*3.55
F. 77	6543 80 6544 0720	467	ALL COACE				and the second
		~ 658	ALL SPACE (%)				50. 30.c
	6548 00 (*********	658 659	RTN		ENERGIA DE CONSTRU	the state of the state of the state of	
i <b>a</b> ndig		660 ****		<b>建筑的大学</b>			A STATE OF
: <sub>6</sub>		5 (A)	CLR_BEDTTOP_		一条 横 変 動作 かいこうしゅう かいしゅうしゅう	memory in E	
	654C FO'	LLT	MAUC E EDAM OF		Posts in the contract of the c	A Marie Conference of the State	`
A Laboration	6540 350005 ASSN 867864		TOYE S FROM		make morning free		C. 15%
	6553 8F78F0						
, A <sup>t</sup> err <sup>al</sup>	6556 00 34	664	RTN		AND THE STATE OF T	SHOW BELLEVIEW	e de Las
	and the second s		The second angle where the second second				
in the second	and the second of the second		and the state of the			and the second of the second	
	en e	e programa e la compania de la comp La compania de la compania del compania del compania de la compania de la compania de la compania del compania de	and the second of the second o	ting the section	Allendaria Majarahan Salah Salah atau		
					The second		L.St
			AND THE RESERVE TO TH				

Company of the control of the contro

**6** 

Ç, 6

6

```
GPL ASSEMBLER 79.150
                                                    10/17/83
                                                                 09:15:56
                                                                               PAGE 18
 666 ****
                    667
                          *************************
        669
                          * RUN OTHER PROGRAMS FROM DISKETTE
            667 *********
670 ***
671 OTHER
   671 OTHER 6557 0720 672 ALL SPACE
   6559 08FF02 673 FMT XPT=2,YPT=2,':* RUN PROGRAM FILE *;'
655C FE0213
655F 2A2052
  655F ZAZUDZ
6562 554E20
6565 50524F
   6562 554E20
6565 50524F
6568 475241
                                                  6568 402046
656E 494C45
6571 202AFB
  6574 B64908 674 SB & &FLAG2.OTHR Set bit for "other" program
  6577 BF2001 676 DST >102.2L0C Put file name? on screen.
  657A 02
     6578 31000A 677 MOVE 10 FROM ROM(#FLN) TO RAM(@LOC) 657E 802062
     6581 58
  6582 A32000 678 DADD >40, QLOC
  6585 40
6586 B64804 679 SB @FLAG.NOTHIN
6589 06640A 680 CALL NAME Get file name input.
681 *
658C 8F5045 682 $IF @FAC6 .DEQ. O THEN Default file name UTIL 1
658F 97
6590:0665B2 683 CALL DOWNLD Call down load program. 1
6593 662D 684 DATA #UTNAME Using DSK1.UTIL1 1
6595 45A6 685 $SELSE 1
6597 0665BF 687 CALL DOWNC2 Prepare PAB area 1
659A 0663C3 688 CALL COPYNM Copy name into PAB area 1
659D 07B04E 689 $ $IF RAM(0FAC4) LOEQ. :CS: THEN Cassette interface 2
  65A6 066504 693 CALL DOWN10 Load 1t.
694 * Following branches to >8300(@TEMP) to execute the program
65A9 066539 695 CALL COLOR2 Color for execution time
65AC OFFO 696 XML >FO Get the first address and branch
65AE 67F7 697 BS ERRXML In case of error
   LAST ERROR FOUND AT LINE 224
      ***** ERROR IN IMMEDIATE FIELD LINE 698 COLUMN 17
    ***** ERROR IN BRANCH ADDRESS LINE= 698
6580 0000 698 BR EX20 End of execution code
                                                   End of execution code
                  700 **********************
                    *************************
                  703 * First 6 bytes of the program file consists of :
```

739 RTN

6613 00

GROM

10/17/83

09:15:56

PAGE 19

GPL ASSEMBLER 79.150

GPL ASSEMBLER 79.150 GROM 10/17/8 6614 050013 740 LDDATA DATA 05.0.#BUF4.#0.#>2100.#10.:DSK1.: 6617 800000 661A 210000 661D 0A4453 6620 4B312E 6623 454449 741 EDNAME DATA :EDIT1: 6626 5431 6628 415353 742 ASNAME DATA :ASSM1: 6628 4D31 6628 4031 6620 555449 743 UTNAME DATA :UTIL1: 

```
745 ****** TITLE ASSEM
              746 *********************
               747 * July 17. 1981 Author 5. Endo
                 748
                          ***********
               749
                     * ASSEMBLER
        9DD2 752 OPTLN EQU >20D2-OFFSET Oution line for assembler
                 753 ******
                 754
                     ASSEM
                 754 ASSEM
755 ALL SPACE
757 FMT XPT=2,YPT=2,':* ASSEMBLER *:'
    6632 0720 755
    6634 08FF.02
                 759
                 760 DCLR &FLAG
    6548 8748
                                              Clear all flags.
                 761 ST >7E.0SUBSTK
    664A BE737E
                                             Reset subroutine stack.
    664D 864820 ..
                 762
                           SB
                               &FLAS.ASRC Now we are in assembler
                763 *
   6650 D78F9D - 764 $1F @IDCODE .DEQ. >AA55 GOTO ASM10 It's already loade 6653 OOAA55
    6656 6687
    6658 0892A0
                 766
                        FMT 194,17,':LOAD ASSEMBLER(Y/N)? ::
                 768
                    CALL YORN Get response, yes or no.

$IF akey .Eq. BACK GOTO START Back key or in: key
   6672 06679F
                 769
    6675 D6750F
    6675 D6750F 770
6678 6079
   6678 6079 770 $1F dREY EQ. BACK GOTO START Back key or :n: key 6678 6079 771 $1F @KEY .EQ. :N: GOTO START will take you back to
  6670 6079
                      ****
6684 D665D4 777 CALL DOWN10 Down load it # 778 **************

778 *************

779 * Give additional soace for assembler - CORY needs this ************

780 *************

781 ASM10 ***********

6687 BFAF13 782 DST >0114.RAM(BUF4)

6688 BF5613 783 DST BUF4.@FAC12

6690 BC

6691 BE4CO4 784* ST 4.@FAC2 Number of files disk DSR handles

6694 D60010 785 CALL DSRLNK Call files

6697 DA 786 DATA 10
                 786 DATA 10
    6697 OA
  6688 00
    66B9 BE2000 794 DST >0C2.aLOC Get loc for source file name.
     668C C2
                      CALL DSR2
    668D 0663A3 795 *
    6600 BFE002 797 DST BUF1.RAM(BUF(SREF)) Source buffer address
```

GPL ASSEMBLER 79.150

```
GPL ASSEMBLER 79.150 GROM
   6603 101080
   805 FMT XPT=2.YPT=8.':0BJECT FILE NAME?:'
807 DST #PAB2.@SREF 0...
   6609 08FF02
                  DST #PAB2,@SREF Put obj PAB address
   66E1 BF1C11
   66E4 00
66E5 BF2001
                  DST >142,8LOC Set file name cointer.
             808
   66E8 42
                              Prepare PAB area
                   CALL DSR2
            809
  % 66E9 0663A3
             810 *
             810 *
811 ST 00,RAM(FLG(SREF)) Open with update mode
   66EC BEE001
   66EF 1000
   66F1 BFE002
             812 DST BUF2, RAM(BUF(SREF)) Buffer address.
   -66F4 1C1180
                    CALL DSR3
   66F7 0663E9 813
                                    Open output file.
            814 SB OFLAG.AOBJ
815 *
  66FA 864840
                                    Flag for obj file open
             aFLAG, NOTHIN
                                   No file name is OK.
Set pointer to name
   6717 B64804 821 SB @FLAG.NOTHIN
671A BF2001 822 DST >1C2.aLOC
 671D C2
671E 0663A3 / 823 ... CALL DSR2 Prepare PAB area.
6721 8F5067 825 ... $IF @FAC6 .DEQ. O GOTO ASKOPT No list file. Skip.
 6739 FE1007
  673C 4F5054
  673C 4F5054
673F 494F4E 6
6742 533FFB
  6745 BF2002 835 DST >242.aL0C Specify cursor location 6748 42
    6749 06640A 836 CALL NAME
674C B248FB 837 RB OFLAG.NOTHIN No input line was CK so far.
    838 * Option is transfered to ERAM
674F 8F5047 839 * SIF 8FAC6 .DEQ: 0 THEN No option. It's OK 1
6752 56
   . 6753 BD4E20 3 840 3 7 7 9 0ST aLoc. aFAC4
   6752 56
```

PAGE 22

10/17/83 09:15:56

GPL	. ASSEMBLE	R 79	150	i di di	GROM	10/17/83	09:15:56	PAGE 23
der B		A. V. Ja						lander († 1865) Lagoriansk fransk
3 3 1	for who will be say.	841	1427.498	\$END	IF The State of th			
6756	35000F	842		MOVE	15 FROM RAM	(aFAC4) TO 80	PTLN Move	to RAM
6759	8F9002				Constitution and a			and the second second
575C	BO4E	Para 13				ing a set of the set o		The second
1.00 N [1]		843 %	***	gili tay.	2574 til 220 bli i 1900 i			
化氯化剂		844	*****	****	Now asse	mble !!!!!!	*****	
Calley 1		845	***	ergest a		Mary of Free par	च्यात्रकातिकातिका <u>ः</u>	
675E	066549	846		CALL	CLRTOP	Clear editor	informatio	n
6761	0720	847	1.5	All	SPACE			
6763	0F21	848		XML	ASM	Assemble the	orogram.	*
6765	67C7	848 849		es T	EMSG	Assemble the Error routin	e shared wi	th editor
1,40		850	*****	**	Close files	before going	back to ma	97U *****
1.0		851	<b>★</b> 1	٠.	grand to		a ·	
		852	ASMOUT					
6767	066549	853		CALL	CLRTOP PAB1.8SREF	Clear e	ditor infor	rmation
676A	BF1C10	854		DST	PAB1.8SREF	Close s	ource file	
	00				$(\mathbf{v}_{i})\cdot \mathbf{v}_{i} = \mathbf{v}_{i} + \mathbf{v}_{i} + \mathbf{v}_{i}$	* * *	4. 4	
676E	0663E5	855		CALL	CLOSE			•
	BF1C11					Close c	bject file	N
	00				,		, -	
	0563E5	857		CALL	CLOSE			• •
A778	DA4880	858		SIF.	.BIT(ALST) a	FLAG .EQ. 1 T	HEN	1
	6784		200	· · · · · ·		FLAG .EQ. 1 T	A sports	Commence of the Section 1
4770	RE1012	859		051	T PARS ASRE	F Close 1	ist file	4
6780	0036366							AN ALL OF THE STORY
A781	066365	860		CAI	L CLOSE			1
	1 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		the second of the second of	40 CT N. L. CS	The state of the s	and the second of the second o		
6784	8648	8.62		CLR	&FLAG	The same of the sa		
and the second	교육 시간 사람들은 그 사내가 되었다.	2 4 2	****	• * * ± 5				
6786	DAA938**	8848		CALL	CONT	Issue messac	e and cont	inue:
6789	84079 W	A A C	are properties?	BRIG	START	Go back to r	enu screen	SERVE WAR WALK
	The Control of the Control of	8448	*****			tal Director of Scotl		
1. 3.54		867	****	4 10 70	and the second s			
	a de la companya de La companya de la co	AAA	*****		THIS TO THE		J. S. North Co.	
	ASSET TO SELECT TO SELECT THE SEL				. A PAPER A 76 TO S. JOHN C.	A CONTRACTOR OF THE	TO SAFERY	AND THE PROPERTY.
200		ัลรถ		2,580,580G		军队建建了一个	44 6 6 6 6 C	CHAIN THE WALL
6780	LELELE TO	877	้าก็ก็พรร	DATA	CONE MOMENT	PLEASE	Challe and the	CONTRACTOR OF THE STATE OF THE
* 0.00	<b>(F) (T) (E) (T)</b>			ryntin Valori	or a serrous - service 14.5 Property - service		1 July 2003 F 20 Y.	

GPL	ASSEMBLE	R 79.	150	GROM	10/17/83	09:15:56	PAGE	24
100						All Control of the State of the		
	i de la propies						**	
				And Africa College of				
		876	***	the countries of the control of the countries of the coun	manager and the second			
٠		6//	****			name tensebasand		
a strong s	New Jan Jan 1	0/0	*****	ger likes or	'no" resconse f	tom vehnoard	er sa l	
e prise projecti. Na september		0 / 7 · · ·	TATATA					
470C	DEAGAE	ପ୍ରଧା : ପ୍ରଧା :	TORN	ST CURSOR.8	LIBRUE			
47A2	8470	883 	a chirac	CLR ATIMER	**** *********************************		A STATE OF THE STA	
67A4	CD1870	884		EX CB, a CHRBU	JF			
		885	The Strike Street	*REPEAT	Wainting fo	r input.	•	
67A7	03	888		SCAN				1
67A8	67B1	887		BS SCANSS	<b>".</b>	•		1
				\$UNTIL aTIMER	.HE. 6			
LTAR	/ T A "							
67AF	47A2	889		BR SCANSO	•			
		890	SCANSS					
67B1				\$IF &KEY .EQ.	BACK GOTO YNEN	D		
			s in the	the second second	•	•		
6786	D67559	892		\$IF &KEY .EQ.	:Y: GOTO YNEND			
6789	67CO -							
			, and a second s	\$IF aKEY .NE.	:N: GOTO SCANE	0		
	47A2						•	
		894	YNEND					4 4 4
67C0	BC7075	895		ST &KEY, CB	Show charac	ter on screet	1.	
67C3	00	896	ر برسو ۱۰۰۰	RTN 1	and the second of the second o	and the second of		
ه المستحد	La Carago C. Villago.	877	***	RTN	and the state of t	San	1 1 4 -14	
4. 34	Andrew La	070	***	Samuel Committee ( 1860)	A SAN PARELLE CONTROL	i capaga jagabar	gagari ka terpad	. Admi
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	ng Lawaran Na	077			The grant of the control of the cont	ta waxi ya Satsiya 27 18 ya s		1500

Mr. 3

```
3,57
                  901 *** TITLE ERROR
                  902
                      ********
                  903
                            July 17, 1981 Author S.Endo
                  904
                             *******
                                               *************
                      * BERROR HANDLING ROUTINES
                  905
                  906
                  907 A. * Caracina and A. S. Caracina and C.
                  908 * All GPL errors are handled in this section
                  909
                      **************************************
                  910
                      * EMSG -- error message retured from assembly language
                  911
                            utility program such as ASSM, EDIT, ALUTIL
                  912
                       ******
                  913 - EMSG1 -
    6704 066525
                  914
                             CALL COLOR
                                             Special entry for editor
                  915
                      EMSG
    6707 066810
                  916
                             CALL FIX
                                             Fix screen and branch BASIC error
                  917
    67CA CA2208
                             $IF BERROOD .L. 8 GOTO ERRIO
   67CD 4871 -
   67CF A62208
                918
                           SUB 8. BERRCOD
    6702 C62205 919
                             $1F BERROOD .H. 5 GOTO RETURN No error
    6705 67E8
6707 0668A7
  CALL CLSALL
921 CASE BERROOD
67DC 4823 922 BR ERRMEN
67DE 481C 923 BR WRNCHR
67EO 483C 924 BR BADTAG
67E2 4843 925 BR
67E4 484A 926
                                              Close all files. No error checks.
                                              Code 8: Memory full
                                           Code 9: Control character
67EO 483C 924 BR BADTAG Code A: Bad tag
67E2 4843 925 BR ERRCSM Code B: Checksum error
67E4 484A 926 BR ERRDDF Code C: Duplicate definition
67E6 4851 927 BR ERRURR Code D: Unresolved references
928 *
929 RETURN
67E8 DA4820 930 $1F .BIT(ASRC) &FLAG .EQ. 1 GOTO ASMOUT
                        LAST ERROR FOUND AT LINE 698

***** ERROR IN IMMEDIATE FIELD LINE= 931 COLUMN= 44
 ***** ERROR IN ADDRESS FIELD LINE= 931
    67ED DA4901 931 $IF .BIT(LD) aFLAG2 .EQ. 1 GOTO LOAD 67FO 6067
 67F0 6067
67F2 0668C4 932 CALU CLSNO
67F5 4133 933 BR, EDIT Go back to editor
1.0分 67F5 14133 高级
                  934 ********************
                  935 * ERRXML Error returned from user's program * 936 * or program file loaded by selection 5 *
                      937
    938 ERRXML
67F7 066810 939 CALL FIX
                         CALL FIX Fix screen and branch to BASIC error CALL COLOR Reload screen
    67FA 066525
                  940
                  941
                          The following error can only be used by user execution
    67FD D6220F
                             $1F BERROOD .EQ. >OF THEN
   : 6800 4809 .
                  943 ERRPR2 CALL ERROR :Program not found:
944: DATA #PRGNF
945 BR ERRNM2 Clear msg. Go back
    6802 0668E4 ...
   6805 6A29
    6807, 485D
                                              Clear mso. Go back for new input. 1
                  946
                             $END 1F
CALL ERROR
    5809 0568E4
                  947
                                             :XML error:
    680C 69BF
                             DATA #XMLERR
```

GPL ASSEMBLER 79.150

09:15:56

PAGE 25

10/17/83

( no.

```
PAGE 26
                        GROM
                                 10/17/83
                                          09:15:56
    GPL ASSEMBLER 79.150
                                         680E 4029 949 BR BEGIN2 Initialize all except top memory
            950 ***
            951 * ****
                     FIX -- Fix screen and branch to BASIC errors
  952 ****
953 FIX
6810 390001 954 MOVE 1 FROM ROM(#ON) TO VDP(1) Turn on the screen
6813 016271
  6810 390001
  ~ 6813 O16271 🤄 -
               $1F .BIT(BSC) aFLAG .EO. 1 GOTO ERRBSC Error in BASIC
  6816 DA4808
   6819 4D9C
   681B 00
            957 **
             959 * General warning routines
            961 WRNCHR
962
             960 ***********************
   ARIC DAARDS
                                :Control character removed:
   681F 69CA
            963
                     DATA #CHRWRN
                  BR RETURN
   6821 47E8
            964
            965
             966 ***
            949 ERRMEM
                                :memory full error:
 6823 0668E4 970 CALL ERROR
6826 6982 971 DATA #MEM
6828 47E8 972 BR RETURN
                                 BR RETURN
           972
  973 *
974 ERRPRG
LAST ERROR FOUND AT LINE 931
  ***** ERROR IN IMMEDIATE FIELD LINE= 989 COLUMN= 17
   ***** ERROR IN BRANCH ADDRESS LINE= 989
   6841 0000 - 989
                 BR LOAD
             990
             991 ERRCSM
   6843 D668E4
                     CALL ERROR
            992
                                :checksum error:
           993
                   DATA #CSMERR
   6846 69F0
   LAST ERROR FOUND AT LINE 989
   **** ERROR IN INMEDIATE FIELD LINE= 994 COLUMN= 17
   ***** ERROR IN BRANCH ADDRESS LINE= 994
   6848 0000
          994
                BR
                        LOAD
```

```
996 ERRDDF
                                                                        PAGE 27
         GPL ASSEMBLER 79.150
                                    GROM
                                                10/17/83
                                                           09:15:56
        684A 0668E4 997 CALL ERROR :duplicate definition:
684D 69FF 998 DATA #DUPDEF
        LAST ERROR FOUND AT LINE 994
        ***** ERROR IN IMMEDIATE FIELD LINE= 999 COLUMN= 17
        ***** ERROR IN BRANCH ADDRESS LINE= 999
684F 0000 999 BR LOAD
1000 *
        1000 *
1001 ERRURR
6851 0668E4 1002 CALL ERROR
6854 6A14 1003 DATA #UNRREF
                                               :unresolved references:
        LAST ERROR FOUND AT LINE 999
        ***** ERROR IN INMEDIATE FIELD LINE= 1004 COLUMN= 17
        ***** ERROR IN BRANCH ADDRESS LINE= 1004
        6856 0000 1004 BR LOAD 1005 *
                   1005
                   1006 ERRNAM
      6858 0668E4 1007 CALL ERROR : Name too lon 6858 69B1 1008 DATA #NAM 1009 ERRNM2 $1F 0FAC6 .DEQ. 0 GOTO ERRNM3
                                              : Name too long:
      6860 6C
6861 BEB04E 1011 ST SPACE, RAM(@FAC4)
6864 20
                                                   Delete name on screen
    6864 20
6865-3450E0 1012 MOVE @FAC6:FROM RAM(@FAC4) TO RAM(1(FAC4))
6868:014EB0
6868-4E
1013 ERRNM3
686C 06694B 1014 CALL CLNMSG Clean up message area

LAST ERROR FOUND AT LINE 1004.

***** ERROR IN IMMEDIATE FIELD LINE= 1015 COLUMN= 17
    ***** ERROR IN IMMEDIATE FIELD LINE= 1015 COLUMN= 17
     ***** ERROR IN BRANCH ADDRESS LINE= 1015
686F 0000 1015 BR EXEC
       1019 ERRIO
     6871 BC4EEO 1020 ST RAM(FLG(SREF)), @FAC4 Get error code out.
6874 011C
       Move 5 bits to right justif
       6870 B64E30 1023 - OR >30,8FAC4
                                                       Add ASCII offset.
                    1024
                           AND >1F,RAM(FLG(SREF))
       687F 82E001 1025
6882 1C1F
                                                       Get error code out
                    1026 *
                   1027 * Next line checks if error is reading after end-of -file.
                    1028 * .
        6884 D64E35 1029 $IF @FAC4 .EQ. >35 THEN In case of just an end-of-file1
        5887 488E
        6889 0668A7 1030 688C 4133 1031 1032 4FN
                                CALL CLSALL
                                                    Close all the files
                            BR EDIT
                                                    Branch back to editor
```

588E U668E4 1U34 CALL ERROR Regular I/O error.
6891 698E 1035 DATA #IO
1036 ERRIO2
5893 U668A7 1U37 CALL CLSALL Close all files without checking err
1038 \*\*\*



```
b..
    6896 DA4820 1039 #1F .BIT(ASRC) @FLAG .EQ. 1 GOTO START
    6899 4079
    LAST ERROR FOUND AT LINE 1015
    ***** ERROR IN IMMEDIATE FIELD LINE= 1040 COLUMN= 40
                                           3.4
     ***** ERROR IN ADDRESS FIELD LINE= 1040
     6898 DA4901 1040 ... #IF. .BIT(LD) 8FLAG2 .EQ. 1 GOTO LOAD
     689E 6079
    68A0 DA4908 1041
                       $IF .BIT(OTHR) &FLAG2 .EQ. 1 GOTO BEGIN
    68A3 4025 13
               1042 8R EDIT
    68A5 4133
                                                      Go back to editor
               1043 ****
                1044 CLSALL
    58A7 BF1C10 1045
                       DST PAB1.@SREF
                                          Close first file
    00 AA66
    68A8 0668C4 1046 CALL CLSNO
68AE BF1C11 1047 DST PAB2.0SREF
                                           Close second file
    6881 00
    6882 0668C4 1048. CALL CLSNO
6885 BF1C12 1049 DST PAB3.8
                        DST PAB3.8SREF
                                          Close third file
    6888 00 ...
  6888 00 CALL CLSNO CIOSE fourth file used in COPY state
  .... 68CO 0668C4 ... 1052 ...... CALL CLSNO ......
68C3 00 1053 RTN

1054 ****

1055 CLSNO

68C4 BD561C 1056 DST & SREF, & FAC12 Pointer to name length
68C4 BD561C 1056 DST @SREF,@FAC12 Pointer to name length
68C7 A35600 1057 DADD PABLEN-1.@FAC12
68CA 09
68CB BEB01C 1058 ST 1,RAM(@SREF) Close file
68CE 01
68CF 060010 1059 CALL DSRLNK Call DSR -- no worry about erro
6802 08 1060 DATA 8
6803 86E009 1061 CLR RAM(NLEN(SREF)) Clear name len to avoid prob wi-
 6806 1c
 6807.00
68E2 48EE 1070 BR WRNENT
              1071
1072
1073
                      * ERROR HANDLING SECTION
               1075
    68E4 066948 1076 CALL CLNMSG Cleanup message are 68E7 310009 1077 MOVE 9 FROM ROM(#ERRM) TO RAM(>2A2)
                      ERROR
                                          Cleanuo message area
     68EA A2A269
     68ED 57
```

10/17/83

GROM

09:15:56

GPL ASSEMBLER 79.150

GPL ASSEMBLER 79.150	GROM	10/17/83	09:15:56	PAGE	29
					X
1078 WENENT	CCTCH DCAC	CETOU MEGGA	CE ANNOESE		•
68EE 884A 1079	reich drac	reich measa	de Apeness.		
68F0 884B 1080 68F2 864C 1081	CLD SEACHS				
00F2 004G 1U01	MAUCH COAM DAY		CAR STATE		
68F4 330001 1082 68F7 400000	MUVE I FROM RUE	Karac) io ara	CTO	Maria da ay a	i de la composición della comp
68F7 4D0000 68FA 4A					
68FB 324CA2 1083			TO RAM(>202)		
68FE C20001	HOYE WING Z TIN	,,,, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	( ) (((() ) ) W ((x) )		•
6901 4A	and the second second				
6902 074A69 1084	\$IF @FAC .DEO.	#IO THEN	If an I/O	error	. 1
6905 8E490C	11. 41.110 10.01	22 T. 20 . 5 . 1 . 1 . 1 . 1 . 1			
6908 BCA2D2 1085	ST @FAC4.RAP		Disolay e	error code	. 1
6908 4E					
	\$END IF				
690C 074A69 1087	SIF OFAC .DEQ.	#XMLERR THEN			1
690F BF493B	and the state of t				•
6912 E72200 1088	DSRL @ERRCOD	. 4	Chande er	ror code	to1
6915 04					
6916 E62304 1089	SRL @ERRCOD-	+1.4			1
6916 E62304 1089 6919 C62209 1090	#IF BERRCOD	H. 7 THEN	Take care	e of A - P	2
691C 4925	and the second second			•	
691E C6220F 1091	\$IF &ERRCO	H. 15 60TO	CONT Number t	coo big so	o 58 €
6921 6938					
6923 A22207 1092	ADD 7, @ERR	COD	The second second		. 2
6921 6938 6923 A22207 1092 1093	SEND IF	the second second			د شیدید در
0720 002307 1074	*IF SEKKCOD+	i H. 9 THEN	Take care	e of A - f	· 2
	Maria de la Ma La maria de la	The state of the s	And the second		100 miles 202
692B C6230F 1095	SIF BERRCO	D+1 .H. 15 GOT	O CONT Skip t	this numbe	<b>≥r</b> ′′ <b>Z</b> ∴\
6938 6930 1096 6930 A22307 1096 1097 6933 A32230 1098					
9 6930 A22307 1096	ADD 7,8ERR	COD+1			
1097	SEND IF		a mine in many registration	on a complete for the second	
6933 A3223U 1098	DADU_>3U3U,&	ERKCOD	and the second second	American Company of the Company of t	1
6936 30 6937 BDA200 1099 693A 22	000	D.M.C. CO.O.	STATE OF THE STATE		
6 4737 BUAZUU 1U77	USI GERRCUD	, KAN ()ZUU)	ON DISCLAY!	ic on sor	reni .
693A 22		or in the Control of			
1101 CONT 6938 310017 1102	MOUE OF EDOM D	OMIMORECEN TO	DAMINOEON	alam da karata da da karata da	
693E A2E269	HUYE 43 FRUN R	VII. WEREDON IV	NAUL VALAN	a de la composição de la La composição de la compo	
6941 60		LORD BANGS OF		Service Art	27.85
1103 SCAN40					
6942.03	CLVN				
6943 4942 1105	BR SCANAD	Wait till 1	nit enter.		
6945 D6750D 1106	\$1F akey .NE.	FTR GOTO SCAN	40		era geraza era Garago
6948 4942	TAN WILL THEME			- 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1	redictor of
694A 00 1107	RTN			- Yorking	M. Say.
1108 ****					1 Chap 1 St. 1 St. 1
1109 CLNMSG				e jaron en	
694B BEA2AO 1110	ST SPACE.RA	M(>2AO) CLEA	N UP MSG AREA	•	
694E 20	-,- · · · - <del>- · · · · ·</del>				•
694F 35005F 1111	MOVE >SF FROM	RAM(>2AO) TO	RAM(>2A1)	•	Vannag.
6952 A2A1A2					
6955 AO	1 (1 m)				
6956 00 1112	RTN	1977		* * .	

```
GPL ASSEMBLER 79.150
                                      GROM
                                                        1113 *
1114 *******
1115 *
                    1115 * DATA AREA
1116 *******

6957 2A2045 1118 ERRM DATA :* ERROR *:

6960 505245 1119 PRESS DATA :PRESS ENTER TO CONTINUE:

6977 2A2057 1120 WRNM DATA :* WARNING *:

6982 0B4045 1121 MEM DATA 11.:MEMORY FULL:

698E 0E492F 1422 10 DATA 11.:MEMORY FULL:
  698E 0E492F 1122
6990 134E4F 1123
                            10 DATA 14.:1/O ERROR CODE:
RAMERR DATA 19.:NO MEMORY EXPANSION:
                            10
  6981 004E41 1124 NAM DATA 13.:NAME TOO LONG:
698F 0A4552 1125 XMLERR DATA 10,:ERROR CODE:
69CA 19434F 1126 CHRNRN DATA 25.:CONTROL CHARACTER REMOVED:
    69E4 0B494C 1127 ILLTAG DATA 11,:ILLEGAL TAG:
    69F0 0E4348 1128
69FF 144455 1129
                            CSMERR DATA 14.: CHECKSUM ERROR:
                            DUPDEF DATA 20,: DUPLICATE DEFINITION:
    6A14 14554E 1130 UNRREF DATA 20.:UNRESOLVED REFERENCE:
    6A29 115052 1131
                            PRGNF DATA 17,:PROGRAM NOT FOUND:
    1132 ******
6A3B 707070 1133 CDATA DATA >70.>70.>70,>70,>70,>70,>70
  6A43 007E42 1134
                                     DATA 00.>7E.>42,>42,>42,>42,>7E,00
    6A4B E0000E 1135
                            VDPRG2 DATA >E0.>0,>0E,>1,06,0,>F5
```

```
TITLE " BASIC"
                  1139 *****************************
                                 Sept 09, 1981 Author S.Endo
                  1140
                1141
                                ******************
             1142
1143
1144
                         * BASIC SUPPORT FUNCTIONS
                         ********************
                                This program takes care of interface to BASIC.
                 1145 *
                                Routines callable from console BASIC are listed 1
                  1146 *
1147 *
                                 in the link list below.
                 1148" ****
                  1147 . ******
              1150 **
                               XML EQUATES
                         ****
                 1151
                                EQU >12
EQU >13
        0012
                  1152
                          CFI
                                                 Convert floating point to integer
                 1153
        0013
                         SYM
                                 EQU >14
      0014 1154 SMB
      0015
                 1155 ASSGNV EQU >15
1156 VPUSH EQU >17
                                                 Assign value to numeric.
        0017
                                                 Push to stack
        0018
                 1157
                         VPOP
                                  EQU >18
                                                Pop off the stack.
                         PEMCH EOU >1B
                                                 Advance one character
        0018
                  1158
                  1159
         1160; ****
                                GPL CALLS
     1161 *
0038 1162 GETSTR EQU >38 GET ENOUGH SPACE FOR STRING
        001A 1163 WARN$$ EQU >1A Warning handling routine 001C 1164 ERR$$ EQU >1C Error handling routine
001C 1164 ERR$$ EQU >1C Error handling routine
1165 ******

1166 *** CPU RAM EQUATES ***

1167 *****

0004 1168 PC EQU >04 Address pointer

008C 1169 BYTE EQU >0C Number of bytes to move

0010 1170 OLDS EQU >10

0012 1171 CNT EQU >12 Counter.

0016 1172 PCODE EQU >16 ERAM address to store parm information
                                             Counter.

ERAM address to store parm information

A->8300 PARAMETER COUNT INCORMATION
         900A 1174 PARM EQU >200A->8300 PARAMETER COUNT INFORMATION
002¢
               1175 *
1176 PGMPTR EQU >2C Program pointer
1177 *
   1177 *
0040 1178 FREPTR EQU >40 Free space pointer
0042 1179 CHAT EQU >42 Character code
006E 1180 VSPTR EQU >6E Value stack pointer
    VSPTR EQU >6E Value stack pointer

1181 ****

0083 1182 COMMA* EQU >83 Token for ","

0086 1183 RPAR$ EQU >86 Token for ")"

0087 1184 LPAR$ EQU >87 Token for "("

0065 1185 STRING EQU >65 String Token for ")"
                 . . . 1186 . **********************
                                LINK LIST -- list of callable subprograms
                  1187
                   1188
                          *******************
   6A52 6A586A 1189
                         LINKO1 DATA #LINKO2.#INIT.4.:INIT:
  6A55 960449
   6A58 4E4954
```

6ASB 6A646A - 1190 LINKO2 DATA #LINKO3, #BLOAD, 4,:LOAD:

20,750

GROM

GPL ASSEMBLER 79.150

09:15:56

10/17/83

PAGE - 31

```
GPL ASSEMBLER 79.150
                                            AASE ECO44C
     6A61 4F4144
     6A64 5A6D6C 1191 LINKO3 DATA #LINKO4.#BLINK.4.:LINK:
     6A67 080440
     6A6A 494E4B
     6A6D 6A7668 1192 LINKO4 DATA #LINKO5, #PEEK, 4, : PEEK:
     6A73 454548
     6476 648068 1193 LINKOS DATA #LINKO6, #PEEKV, 5, : PEEKV:
     6A79 7E0550
     6A7C 454548
     6A7F 56
                      LINKOS DATA #LINKO7.#POKEV.S.:POKEV:
     6A80 6A8A6A 1194
     6A83 E70550
     6A86 4F4B45
     6A89 56
     6A8A 000060 1195 LINKO7 DATA #0.#GETCHR,7,:CHARPAT:
    - 6A8D 120743
     6A90 484152
     6A93 504154
                  1194
                            INIT -- initialization routine
                  1197
                  1178 *******************************
    1199 INIT
6A96 B64808 1200 SB GFLAG, BSC Set BASIC flag on.
6A99 A32C00 1201 DADD 5.0PGMPTR Skip name len and name 'INIT'
6A9C US
6A9D U66AAA 12U2 CALL INIT3 Force UTIL load
12U3 *
6AAU 40U1 12U4 BR RTBSC2 Go back to BASIC interpreter
12US ***
    6A9C 05
        1206 * Routine to load UTIL subroutines
1207 ***

1208 INIT2

6AA2 D78F9D 1209 $IF @IDCODE .DNE. >A55A THEN Check if UTIL is ther

6AA5 00A55A

6AA8 6AD1
    1210 INIT3

6AAA 066504 1211 CALL CHKRAM IS E-RAM there?
6AAD BE4AO3 1212 ST 3.8FAC Force UTIL load
   6AAD BE4AO3 1212 ST 3.8FAC Force UTIL load 1
6ABO BF4C70 1213 DST ALCODE.8FAC2 1
6AB3 00 $
6AB4 1214 INITLP EQU $
     6AB4 330004 1215 MOVE 4 FROM ROM(@FAC2) TO @FAC4 Load utility routine1
    6AB7 4E0000
6ABA 4C
6ABB A34C00 1216 DADD 4.@FAC2
6ABE 04
6ABF 324ECF 1217 MOVE @FAC4 FROM ROM(@FAC2) TO @->8300(FAC6) 1
                            6AC8 A14C4E 1218
                             DADD @FAC4.@FAC2
     6ACB 924A 1219
                 1219 DEC OFAC
1220 BR INITLP
     6ACD 4AB4
6ACF 4549
                 1221 BR
                                  CLRTOP ---
                                               Clear editor information and return1
```

10/17/83

09:15:56

PAGE

GPL ASSEMBLER 79.150				PAGE 33	3
				,	
1222 \$END	$-\mathbf{IF}$ (Left) $p_{\mathbf{x}'}$ ( $p_{\mathbf{x}'}$ ) $p_{\mathbf{x}'}$ ( $p_{\mathbf{x}'}$		and the second of the second		
6AD1 00 1223 RTN		and the second	All processing and the second		
6AD1 00 1223 RTN 1224 *** 1225 ****					
1226 SKIPN 1				ar de la	
6AD2 864A 1227 CLR	2FAC	Count size o	of routine name		. 44
-6AD4 BC4880 21228 5 ST	.RAM(@PGMPTR),@F	AC+1 Copy s	ize 🦠 💮	Buditak a	
6AD7 2C	The second secon	was a sure with the second			
6AD8 A12C4A 1229 DADD					
6ADB 912C 1230 DINC		,			æ
6ADD 0F18 1231 XML	PGMCH	. Get next tol	(en		
6ADF D642B7 1232 \$IF				<b>p</b>	
6AE2 4E07			in the state of th		
6AE4 OF1B 1233 XML	PGMCH	Advance poi	nter one more	time	
6AE6 00 1234 RTN		. •			

1,

```
GROM
                                            09:15:56
             1236 ***
                      1237 *****************************
            1240 POKEV
   6AE7 864801 1241 SB @FLAG, VFLG Turn on VDP flag
6AEA 4AF1 1242 BR LDP1 Join LOAD routing
             1243 BLOAD ...
   6AEC 8748
             1244 DCLR &FLAG
                                   Delete all flags
             1245 SB OFLAG. BSC
1246 *****
   6AEE 864808
             1245
                                   Called from BASIC program.
             1247 % LDP1
                                  Skio name first
   AAE1 DAAAD2
             1248 CALL SKIPN
             1249 LDP3
   SAF4 DEBS
             1250 PARSE RPAR$ Parse up to ")".
             1251 ** Process file name if it is a string.
1252 *IF @FAC2 .EO. STRING GOTO LDP2
    6AF6 064C65
                    $1F @FAC2 .EQ. STRING GOTO LDP2
   6AF9 6B30
             1253: ** Otherwise, it is an absolute address
   ,6AFB OF12
             1254
                     XML CFI
                               Convert to integer
    6AFD 065403 1255
                     $IF @FAC10 .EQ. 03 GOTO ERRNTB
                                              Check for overflow
    6800 6E18
   6800 6818
6802 BD044A 1256 DST @FAC.@PC Save in ERAM(or VRAM) address pointer
1257 LDP4
6805 D642B3 1258 $IF @CHAT .NE. COMMA$ GOTO RTBSC RETURN TO BASIC.
  6808 4CFC
680A 0F1B 1259 XML PGMCH Skip ","
680C 0E86 1260 PARSE RPAR$ Parse up to ")"
680E 064C65 1261 $IF @FAC2 .EQ. STRING THEN NO more data?
                            6813 8F5068 1262 $IF @FAC6 .DEQ. 0 GOTO BEXTLD Null string?
 6B1E 6E1B
  1268 ***
  682E 00
   [6837]80]
    6838 9104 1275
                    6838 9104 1276 DINC aPC 683A 056805 1277 B 683 LDP4
                                        Increment ERAM address.
                                    Continue loop
             1278 *****
                         1279 LDP2
```

The state of the s

GPL ASSEMBLER 79.150

**Figure** 

\*

10/17/83

PAGE 34

```
-.vn
              6830 8F506B 1280 $ $1F 8FAC6 .DEQ. 0 GOTO BEXTLD Skip null string.
   6840 75
  6841 DA4801 1281 $IF .BIT(VFLG) @FLAG .EQ. 1 GOTO ERRSNM Not for VDP poke
   6844 4E20
             1283
                      Following section prepares PAB area for object file
        1284
  6846 BDOCSO 1285
6849 A30COO 1286
684C 5A
                     DST @FAC6.@BYTE
                                          Store actual spec. length
                      DADD PABLEN+80.0BYTE Add in the PAB len. and buffer len
                    XML VPUSH
CALL INIT2
CALL GETSTR
  6840 0F17 1287
684F 066AA2 1288
6852 060038 1289
                                         Push possible temp. string
                                          make sure Utility is there and try to allocate space
   6855 OF18
              1290
                      XML VPOP
                                          Restore original string data
              1291
              1292
                       Variable specification
              1293 - *
                      FAC+4.5 Start address of original device spec.
              1294 *
1295 *
                       FAC+6,7 LENGTH OF ORIGINAL DEVICE SPECIFICATIO
                       SREF
                                Location of PAB in VDP memory
                              Length of entire PAB, including spec.
              1296 *
              1297 *
  6857 0663FE 1298
685A 0663C3 1299
                     CALL CLRPAB Clear the entire PAB
CALL COPYNM Copy device name and length
  6850 BEE008 1300 ST >60,RAM(SCR(SREF)) Screen offset 6860 1060
   6860 1060
   6862 BEE001 1301
  ST &00000100, RAM(FLG(SREF)) Disp, fixed, seq, input
686E BDE002 1304 DST @FAC6; RAM(BUF(SREF)) Store buffer address in PAB
  6873 0000 1305 BR BENTLD Enter object file
1306 ****
1307 ********
   1311 *
6875 D64283 1312
  6875 D64283 1312 $IF &CHAT .NE. COMMA$ GOTO RTBSC Go back to BASIC.
6878 4CFC
687A 0F1B 1313 XML PGMCH (SKIP COMMA)
687C 4AF4 1314 8R LDP3 (CONTINUE IN MAIN
                                       (SKIP COMMA)
                                                (CONTINUE IN MAIN LOOP)
             1315 *******
```

GROM

PAGE 35

09:15:56

10/17/83

GPL ASSEMBLER 79.150

```
1318 *******************************
          1319 ** PEEK AND PEEKV ROUTINE
           1324 * CALL PEEKV (Address, numeric variable....)
              1326 * RETURNS THE VALUE AT address IN ERAM INTO
            1327 * numeric-variable. IF MORE THAN ONE numeric-variable IS
            1328 * SPECIFIED THEN address IS INCREMENTED AND THE VALUE
                    IN ERAM AT THE NEW address IS ASSIGNED TO THE NEXT
              1330 * VARIABLE AND SO ON.
1331 *
1332 PEEKV
   687E B64801 1333 SB OFLAG.VFLG VDP FLAG SET
6881 4885 1334 BR PEEKO ENTRY POINT
1335 PEEK
                   DCLR @FLAG
             1336
   AB83-8748
                                 CLEAR FLAG
   ... 1337 PEEKO ... 6885 066AD2 1338 CALL SKIPN
                             SKIP NAME FIRST
  1339 * CALL INIT2 CHECK UTIL IS THERE AND LOAD
1340 PEEK1
6888 DEB6 1341 PARSE RPAR* GET VALUE OF ADDRESS
   688A D64C65 1342 $1E 8FAC2 .EQ. STRING GOTO PEEK3 SKIP NULL STRING
  6880 68AC
688F 0F12 1343 XML CFI [CONVERT FAC TO INTEGER]
6891 065403 1344 #IF 8FAC10 [EQ. 03 GOTO ERRNTB COVERFLOW?]
6B9C 4CFC.
1347 *
1348 PEEK2:
689E OF18: 1349: XML PGMCH SKIP "," 6
68AO D64267: 1350 : $1F achat .eq. >c7 Then String 3
68A3 48B9
                                                     6BAF 20
6880 D64283 1355 $1F &CHAT .NE. COMMA$ GOTO RTBSC RETURN TO BASIC 2
   68B3 4CFC
   6BB3 4CFC
6BB5 0F1B 1356 XML PGMCH SKIP COMMA 2
6BB7 4BB8 1357 BR PEEK1 GO BACK TO THE LOOP 2
1358 $END IF
   1359 $END IF
6889 CA4280 1360 $IF achar .HE. >80 GOTO RTBSC GO BACK TO BASIC?
   6BBC 6CFC.
6BBC 0F13 1361 XML SYM GET SYMBOL NAME
6BCO 0F14 1362 XML SMB GET VALUE POINTER
6BCO 0F17 1363 XML VPUSH SAVE FAC AREA
```

GROM

10/17/83

119:15:5A

GPL ASSEMBLER 79.150

GPL ASSEMB	LER 79.15	o GROM	10/17/83	09:15:56	PAGE	37
	11 1 1 ×		医马克二氏腺素 医二苯基			
					100	
	1364 *			a cargott father		
	1365 **	GET VALUE FROM ERAI	1 OR RAM INTO F	AC+1		
	1366 *					
68C4 DA4801	1367	\$1F .BIT(VFLG) &FLAG	EQ. 1 THEN			1
ARCY ARCE	and the second process.					200
ABCO BCENBO	1368	ST RAM(QPC), QARG	H1 GET VAL	UE FROM VDP R	AM .	1
6BCC 04	Tare to the contract of	The Section of the Control of the Co	of the conformation at the first			
68CD 48DC	4749	\$SELSE DSUB OFFSET.aPC	GET VAL	HE FROM CPU R	AM	1
68CF A70483	4 7 7 7	DOND ACCOUNT ARC	TAKE OL	T DEESET		1
6802 00	1310	0300 011361.016		.:		•
- 0002 - 0000000 - 7007 - 0000000	4774	ST &O(PC), WARG+	C MOUS UA	HE FROM FRAM		1
6806 GCC4	13/1	SI COLECTIONED.	I HOVE YE	COL TRUIT ERRIE	1.5	*
4504 430467	4770	DADD OFFSET, aPC				1
	1372	DAUD OFFSEI, WFC		* *		•
6608 OO						
		SENO IF	* * *.			
	1374 *		። መነ ምጣ መደ መልጆች የተረጠ ፣	HTO: EAC ABEA		
		* CONVERT INTEGE	K IO LEDATING I	NIU PAG AKEA		· ·
	1376 *	244	CI TAB C	"ላይ ልውጠል መጀመሮች		
	15//	CALL CLRFAC	CLEAR F	TO HICH TINDS		, F
6BDF 8ESD4B	13/8	\$IF BARG+1 .EQ. O G	OLO PRIF IF 76	INU		
69E2 FC		أأحد فالمتعاد المتعاد	.:	n merker kann		,
		ST >40.2FAC	IF LESS	3 THAN 100	DATION	
68E6 865C	1380	CLR BARG	PREPARE	FOR WORD OPE	RATION	J. 39 May 1
6BE8 AESC64	1381	DIV 100.8ARG ST 9ARG.8FAC+1 ST 8ARG+1,8FAC+2 \$IF 8FAC+1 .EQ. 0 T	DIVIDE	BY 1UU	<b>张利亚尔维斯</b>	
- ABEB BC4B5C	1382	ST : DARG. OFAC+1 :4	VALUE .	IF OVER TOU		
SBEE BC4CSD	1383	ST &ARG+1, &FAC+2	nove to	the third di	git.	
68F1 8E4848	1384	SIF OFAC+1 .EQ. U I	HEN VALUE :	is less than l	uu .	Security of the second
68F4 FA				A Rock		
6BF5 CO4C4B	1385	EX 0FAC+1,0FAC+ \$SELSE INC 0FAC	2 Move ti	ird byte to s	econo	7 74 <b>1</b> 7844
6BF8 4BFC	1386	SELSE VERNER OF THE	Market If over	: 100 - 100		1 47 W
- 68FA) 904A 🐃	1387	INC OFAC	Increme	ent exponent v	alue 🦠 🧢	
eth (Cold State State) Rose Rose State Stat	1388	SEND IF	The state of the s			100 C
	1389 *	and the second s				
	; 1390 ; *	KIP	and the same of	en anderstand the second s	190 125 190 180	
	"1391≒°S	KIP,				
6BFC OF15	1392	XML ASSGNV (ASS	IGN TO NUMERIC	-VARIABLE		
		SIF ACHAT .NE. COMM	A\$ GOTO RTBSC	en restrict for careful at a		
6001 4CFC	Ser E.					
6003 9104	1394	DINC &PC (INC POI	NTER TO NEXT E	RAM_ADDRESS}		the state of the s
605 05689E	1395	B PEEK2			AL AND SOL	and the second s
					7. 7. P. E.	
	ii 1397 ∵ *	****				
	را ج در به و محمد این	en e	Company of the Compan	The second s		The bound with the second
			A Committee of the Comm			<b>高等依守东门</b> 。
						Alexandra Company
The second section is a second	The second second second	a stronger i strong i skiller i kalendari	64일 이번 시험 스타웨스트 관계		and the property of the page	

The property of the second second

0

(

9

22

8

•

6

0

100

(4)

(3)

```
1399
               1400
               1401
                        LINK INSTRUCTION: SE September, 1980
               1402
             1403 **************
            1404
         分类 5 1405
           1406
              1407 *
                         CALL LINK("file-name", parameter1, parameter2,....)
             1408
                         LINK ROUTINE READS THE FILE NAME SPECIFIED BY THE USER
             1410
                         AND SAVE THE ADDRESS OF THE NAME FOR LATER USE.
               1411
                         THE FILE WILL BE SEARCHED IN UTILITY CODE LATER ON.
               1417
                         PARAMETERS ARE PASSED EITHER BY REFERENCE OR BY VALUE.
                         NUMERIC OR STRING VARIABLES AND NUMERIC OR STRING ARRAYS
               1414
               1415
                         ARE PASSED BY REFERENCE AND ALL OTHERS INCLUDING A USER
               1416
                         DEFINED FUNCTION ARE PASSED BY VALUE.
               1417
               1418
                         PARAMETER INFORMATION IS STORED IN ERAM >200A THRU >201A
                         THAT GIVES A PARAMETER TYPE CODE OF EACH PARAMETER.
               1419
               1420
                           CODE O .... Numeric expression
                           CODE 1 .... String expression CODE 2 .... Numeric variable
               1421
                         CODE 3 .... String variable CODE 4
               1422
        1423 + *
       1424
                           CODE 4 .... Numeric array
    1425
                         CODE 5 ... String array
 1426
1427
                        IF A PARAMETER IS PASSED AS A NUMERIC EXPRESSION ITS ACTUAL VALUE GETS PUSHDED INTO THE VALUE STACK.
  1427
1428
        1429
                         IN CASE OF A STRING EXPRESSION, ITS VALUE STACK CONTAINS
     1430
                         AN ID(>65). POINTER TO THE VALUE SPACE AND ITS LENGTH.
IF A PARAMETER GETS PASSED AS A REFERENCE THE PRODUCT
173.
1432
1432 * OF XML SYM AND XML SMB IN THE BFAC AREA GETS PUSHED INTO

1433 * STACK.

1434 *

1435 * AFTER AN ASSEMBLY LANGUAGE SUBPROGRAM IS EXECUTED LINK

1436 * ROUTINE WILL POR THE STACK TO CET CO. OF CO.
               1431
   1436 *
1437 *
1438 *
                         ROUTINE WILL POP THE STACK TO GET RID OF PARAMETER INFORMATION. CONTROL WILL BE TRANSFERED TO THE BASIC
 1438 * MAIN PROGRAM AFTERWARDS.
1439 *
1440 **** DATA AREA ****
1441 * CPU RAM FREE SPACE USED IN THIS ROUTINE
1442 *
1443 * FLAG BITS STORED IN LOCATIONS DD TO >OF
                         MAIN PROGRAM AFTERWARDS.
1444 ****
1445 ****
```

10/17/83

09:15:54

GROM

GPL ASSEMBLER 79.150

```
1447 **************************
                1448 * CALL LINK program * 1449 ***************************
 1450 *

1451 BLINK

6008 B64808 1452 SB &FLAG.BSC CALLED FROM BASIC PROGRAM.
600B 066AD2 1453 CALL SKIPN SKIP NAME FIRST
600E 066504 1454 CALL CHKRAM Make sure E-RAM is there
                1450 *
  6CDE 066504 1454 CALL CHKRAM Make sure E-RAM is there 6C11 078F90 1455 $1F 010CODE .DNE. >ASSA GOTO ERRPN No program
                1455
  SC14 DDASSA
1. 6C17, 4E02
  6C17 864808 1456
6C1C 8D106E 1457
6C1F 0E86 1458
                                              Set bit for BASIC flag
                         SB : &FLAG, BSC
DST avsptr.aolds
                                               Save VSPTR for later use.
                         PARSE RPARS
                                               Get the routine name.
   1459 *
6C21 D64C65 1460
                                               Read up to ")",
                         $1F @FAC2 .NE. STRING GOTO ERRSNM Should be a string.
   6024 4E20 -
   6C26 C65106 1461 | #IF @FAC6+1 .H. 6 GOTO ERRBA | Should be less than 6 char
   6C29 6E25
                         XML VPUSH
   602B OF17
                1462
                                               Push to make it semi-permanent.
   602D 8612 1463
602F 8F169D 1464
                1463
                         CLR acnt
                                               initialize parameter counter.
                         DST PARM. &PCODE
                                               Set initial address for parmaeter
   6032 OA .
                1465 *
                1466
                      ****************
                1467
                      * PARAMETERS get avaluated here
 1468
                       *************
                1469 PARC1
 6C33 D642B6 1470 *IF aCHAT .EQ RPAR* GOTO EXED1 No arg. So execute it.
6036 6005
6038 064283 1471
                                                         $1F aCHAT .NE. COMMA GOTO ERRIS Should have a comma.
- 6C3B 4E07
                        1472 *
6C30 BD222C 1473
6C40 DF1B 1474
                         DST @PGMPTR.@ERRCOD Save text pointer.

XML PGMCH Get the character

$1F.@CHAT..HE. >80 GOTO VALO1 Must be an expression.
6C42 CA4280 1475 $1F.
                               1476 * $IF &CHAT .EQ. LPAR$ Pass by expression.
6C47 O66DOA 1477 CALL CLRFAC Clear FAC entry for
                         CALL CLRFAC Clear FAC entry for SYM.

XML SYM Read in the symbol table info.
                                            Read in the sympor table
   604A 0F13 1478
 14798 *
               1480 ×
                         After XML SYM OFAC area contains a pointer to
             32. 1481 € *
                         symbol table.
             1482 *
 1483 *
6C4C DABO4A 1484
6C4F 404C98
                         Below statement checks if it is a UDF.
                         Selow statement Greeks ... 1 GOTO VALO1 Pass by Value.
                                                                         vaiue.
  6052 064283 1485
                                                           Pass by reference.
                       $ $1F aCHAT .EQ. COMMAS GOTO REFO1
 36055 6081 (%)
   6C57, D642B6 | 1486
                         *IF &CHAT .EO. RPAR$ GOTO REFU1
                                                              Pass by reference.
  6C5A 6CB1
6C5C 0642B7 1487
                         $IF @CHAT .EQ. LPAR$ GOTO ARRAY An array.
   6C5F 6C68
   6C61 CA4280 1488 --
                         $IF aCHAT .HE. >80 GOTO VALO1
                                                              Pass by value.
```

10/17/83

09:15:56

PAGE

the state of the s

GPL ASSEMBLER 79.150

6064 6098

```
GPL ASSEMBLER 79.150
                                GROM
                                            10/17/83
                                                                 . PAGE 40
                                                        09:15:56
    6066 4E07 1489 3 BR ERRIS
              1490 *
                1491 ***********************
             1492 * ARRAY case gets checked here
                     **************
   1494 * should look like A(,.) etc.
1495 * Stack entry for an arry will look like
              1495
              1500 * FAC----
                                      ---FAC2-----FAC4------FAC6---
               1501
              1502 *
1503 ARRAY
  1503 ARRAY.
6068 DF18 1504 XML PGMCH Get the next character.
606A D642B6 1505 $IF OCHAT .EQ. RPAR$ GOTO ARRAY2 Pass by reference.
    6060 607B
 606F D64283 1506 SIF 2CHAT .EQ. COMMAS GOTO ARRAY
                                                         More array informatio
   6072 6088 1507
6074 9320 1507
   6074 9320 1507
6076 BE42B7 1508 ST LPAR$.80
6079 40B1 1509 BR REF01
1510 *
                        DUEC APGMPTR
                                                         Adjust the pointer.
                      ST LPARS, OCHAT
                                                         Pass by reference.
 1510 *
1511 ARRAY2
                1512 *
1512 *
1513 * In array cases the symbol table address gets stored
1514 * at FAC area, and the pointer to the value space
1515 * (dimension info.) goes into FAC4
1516 *
2 6C7B OF1B 1517 XML PGMCH Advance the program pointer:
                       In array cases the symbol table address gets stored at FAC area, and the pointer to the value space
6C7D DABO4A 1518 $1F .BIT7 RAM(OFAC) .NE. 1 THEN Test string bit. 1
6C92 A34E00 1527 DADD 6.8FAC4 Point to value space
  6C96 4CC8 1529 BR PUSH
                                            1530 *
               1531
                1532 - * VALUE .
             1533 *
                       Passing the parameter by value
   1534 ******
1535 VAL01
                      ****************
   6C98 BD2C22 1536 DST &ERRCOD, &PGMPTR Restore program pointer.
6C98 OF18 1537 XML PGMCH Skip the first character.
  6098 OF18
                                           Skip the first character.
```

Č.

tin.

```
6090 DEB6 | 111538 PARSE RPARS
                              farsing up to comma [ ]
1540 * After parsing &FAC area contains 1541 * its actual numeric value in a num
               its actual numeric value in a numeric case, and the
        1542 *
               following information in a string case. A SAME --
         1543
        1547
             * FAC-----FAC2-------FAC4------FAC6-------
        1548
   1549
609F 064065 1550
              $IF @FAC2 .EQ. STRING THEN If it is a string
5CA2 4CAB
6CA4 BECOOO 1551
                ST 1,20(PCODE) Store flag for string expression. 1
6CA7 1601
6CA9 4CAF 1552 $SELSE 6CAB 86C000 1553 CLR 80
                CLR ad(PCODE)
                             Otherwise it is a numeric express
6CAE 16
         1554
               $END 1F
6CAF 4CC8
         1555
               BR PUSH
                              Push into stack.
         1556
         1557
         1558
         1559 *
               Passing the parameter by reference *
         1560 ********************
         1561__
              * Variables, array element and whole array passing.
         .1562 ∴ ⊁
     1562 *
1563 *
1563 *
1564 * After SMB &FAC Entry should look like 1565; *
 1579 XML SMB
                              Get the location.
6CB3 CA4288 1580
               $1F OCHAT .HE. >88 GOTO VALO1 Pass array expression.
6CB6 6C98
6088 C3 .
6CBC BEC000 1582
                ST 2,20(PCODE)
                             Must be a numeric variable.
6CBF 1602
         1583 $SELSE
6001 4008
6003 BEC000 1584
                ST 3,80(PCODE) Must be a string variable.
6006 1603
         1585 $ENU IF
```

GROM

10/17/83

PAGE 41

GPL ASSEMBLER 79,150

```
GPL ASSEMBLER 79.150 ...
                                                           GROM
                                                                                10/17/83 09:15:56
                          1586 *
                            * Pushes &FAC entry into a value stack*
                           1590 ***********************
                            1591 PUSH
  6008 9012 1592 INC BONT 600A C61210 1593 $1F BONT .H. 16 GC
                                                                               Increment number of parameter.
                                         sif acht .H. 16 GOTO ERRBA Too many parameters.
      6CCD 6E25
                                             DINC &PCODE
                                                                                Increment parm info address
    6CCF 9116
  1595
6CD1 OF17 1596
                                          XML VPUSH
                                                                                Get the next arqument.
      6003 4033
                           1597
                                             BR PARGI
                             1598
                                       *************
                            4599
                           .1600 * EXECUTE routine
                                        * Restore file name info and transfer *
                         4601
                             1602 * control over to ALC
                            ***********************************
    1604 EXE01
6CD5 BE4A20 1605 ST SPACE, 2FAC Store
6C08 350005 1606 NOVE 5 FROM 8FAC TO 8FAC+1
                                                                                Store blank in the FAC area.
      6CDB 4B4A
     6000 350004 1607
6000 000000
                                           MOVE 4 FROM RAM(12(OLDS)) TO STEMP
       6CE3 10.
LAST ERROR FOUND AT LINE 1305
                                                               The control of the second of t
  ***** ERROR INCIMMEDIATE FIELD LINE= 1608 COLUMN= 33
                                    ***** ERROR IN ADDRESS FIELD LINE= 1608
      6CE4 8F0240 1608 $1F aTEMP+2 .DEQ. 0 GOTO EX10 Do not move 0 bytes.
     6CES 340244 1609 MOVE STEMP+2 FROM RAMISTEMP) TO SFAC Move name to FAC.
     ACEB BOOD
   4CED BD5002 1610 DST aTEMP+2. aFAC6 Copy name length
       LAST ERROR FOUND AT LINE 1608
     ***** ERROR IN IMMEDIATE FIELD LINE= 1611 COLUMN= 13
 ***** ERROR IN BRANCH ADDRESS LINE= 1611
       6CFO 0000 1611 BR EX10 Join execution code.
        1613
                                        1614
                          1615
                              1616 NOERR
       6CF2 CS6E10 1617
                                        $WHILE AVSPTR .DH. BOLDS
       ACF5 4CFC
                                                                                        . Pop the stack.
        6CF7 OF18 1618
6CF9 O56CF2 1619
                                                YML VPOP
                                         ≢END WHILE
                                                                                           Keep popping till no stack for
                              1620
                                        RIESC
                              1621
                                         $IF @CHAT .NE. RPAR$ GOTO ERRIS Check ")".
        6CFC D642B6 1622
        6CFF 4E07
                              1623 RTBSC2
                              1624 XML PGMCH
                                                                                            Advance to the next character.
        6003 8E424E 1625
                                             $1f achar .NE. 80 GOTO ERRIS Check end of statement.
        6006 07
  AMOT 0A0012 1826 CALL BIL Go DEON to BASIC
```

GPL ASSEMBLER 79.150 GROM 10/17/83 1629 1630 RFAC CLROSFACO TOMOS CATALON 1631 CL 600A 864A 1632 600C 350007 1633 600F 484A 6011 00 1634 1631 MOVE 7 FROM OFAC TO OFAC+1 6011 00 1635 Company of the 

9

(47)

```
1637
     1638
1639
     1640
1641
                      * CHARPAT ROUTINE
      1642
1643
            1644 * FORMAT:
           1645 -* CALL CHARPAT open ( < numeric expression> comma
                1646
                                       <string expression> )* close
          1647 * FUNCTION:

1648 * RETURNS THE CHARACTER DEFINTION PATTERN FOR CHARCTER
1649 * NUMBER < numeric expression> INTO (string expression>.
               1650
               1651
              1652 TBLPTR EQU >10 (able pointer 1653 STRPTR EQU >12 String pointer
        0010
                      TBLPTR EQU >10 Table pointer
        0012
               1655
                      GETCHR
  6012 066A02 1656
                       CALL SKIPN
                                     Skio subroutine name
   1657
6015 0E86 145°
                      GCHR2
                1658 PARSE RPAR®
  6017 064665 1659 - $IF @FAC2 .EQ. STRING GOTO ERRSNM Can't be a string
  601A 6E20
 601C 0F12 1660 XML CFI Convert FAC to integer
               1661 * Note that 32 - 159 char are supported in Console BASIC
 601E 065403 1662 $IF &FAC10 .EQ. 3 GOTO ERRNTB
6021 6E18
6023 034A00 1663 $IF &FAC .DLT. #32 GOTO ERRBA
                     $1F @FAC10 .EQ. 3 GOTO ERRNTB (range 32 -> 159)
 6027 CF4A00. 1664 ...
602C.9F6E25
602F E34A00. 1665 DSLL @FAC,3 {8 bytes / entry so multiple, 6032 03
6032 BF1003 1666 DST >300,@TBLPTR {Base of character table less 32*8 }
6037 A1104A 1667 DADD @FAC,@TBLPTR (Add in arg offset)
1668
603A BFOCOO 1669 DST 16,@BYTE Get a 16 byte string
                                           Get a 16 byte string in string space
    6030.10
  1673 **
1674 FFT
   1674 GETLP
6047 BC00B0 1675 ST RAN(ATBLPTR).ATEMP
                                                       604A 10
  6048 E70000 1676 DSRL STEMP, 4
                                                     Shift 4 bits to right.
    604E 04"
    6D4F A20030 1677
                          ADD >30, @TEMP
                                                     Add ASCII "0"
   6D52 CE0039 1678
                          $IF aTEMP .LE. >39 GOTO GCHR3 >39=ASCII("9")
   - 6055 405A
    6057 A20007 1679
                          ADD 7, @TEMP
                                                     Value "A" -> "F"
                1680 GCHR3 .
    6D5A E60104 1681 SRL @TEMP+1.4
                                                    Shift 4 bits for next byte.
```

10/17/83

09:15:56

PAGE: 44

GPL ASSEMBLER 79.150

£6.

* *		200						the second of the second	
- GPL	ASSEMBL	ER 79.15	0	GROM	10/1	7/83	09:15:56	PAGE	45
								The second second	di ar -
er van de skal					en de la companya de La companya de la co		ACCT UMB		
6050.	AZU13U	1682	AUU 23U	l, d FEMT+3		о сонах По сонах	ASCII "O"		
		1683	\$IF aTE	inrti .Lt	737 901	O GCHR4			
6065 4065	4068	4 L D L	ADD 7,8	STEMBAA		Unti	ie "A" -> "F	14.	
			HR4			VAIU		J	
4048	808012	1003 00	ner a	FEMP PAK	CATERTES	New	out in RAM	ta a second	1 4 5
ADAB:	nn	1000	20.	· Citt i Ment	(40111111111111111111111111111111111111		100 III	and the state of	
404C	9110	1487	DINC	TEL PTR	•	Next	. byte		Tar e i
ADAE	9512	1488	BINCT	STRPTR		Move	pointer tw	o bytes	
5070	9204	1689	DINC & BINCT & DEC	TEMP4			it down		
6072	4047	1690	BR (	SETLP		Cont	inue loop -		
1 1 1		1691 *	transfer to	* *	1.0			*	
- f - f -				Y THE ST	RING JUST	CREATE	TO THE STR	RING	
		1693 *	VARIABLE F	FOLLOWING	ž .			1	
±3.		1694 **	*	~		*	e way	***	
		1695 - **	A CONTRACTOR OF THE CONTRACTOR					***	
6074	OF18 :	1696	XML PGMC	4 (SI	(IP.",")			Mark Control	-
		1697	SIF OCHAT	.HE. >8	D GOTO ERS	RIS (D)	DN'T ALLOW T	OKEN}	
	6E07							e e	
	0F13	1698	XML SYM		ET SYMBOL				
			XML SMB						
6075				H (5)	AVE FAC DE	SIACK	FOR ASSGNV)	r Tananan meneralah	
1001		1701 **		سوديس بسوري	DINE CATA	cascas			
		1702	PIP CFAUZ	ME. DI	KING GOIO	ERRSNII	Must be a	scring v	ar.
	, , , , , , , , , , , , , , , , , , , ,	V	ner sons	C 5EAC	Tomp et	a filmer eren	use SREF as		
6089	and the second second	1100	031 7001	Ciarno	· imai b b c i	ring su	use once as	s auuress	1.390
4084	BUZE1C	1704	DST @SRE	F SFACA	Painter	to str	ing 🖟 🤲		8/9
4080	EFENDO	1705	DST 16.8	FACA	String	lenath"		THE TRUE OF	Marian Cons
4000	10 6	AND SEAR SEA		receive ( )	Seld Company	r than the action	STATE OF THE RESIDENCE OF THE PARTY OF THE P	are a come en	
and the second of the	or a second of the contract of	·一定在大学等一次企業時,有一年1800年出版。	Professional and the second of	NV -	Assian	to str	ing variable		e de la company
							Finish loop		
1 5 50 1		A same transfer and the	the second of th		# 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		<ul> <li>A HMMSSE, 17 BW W.</li> </ul>	はからな リントスを行びまでとかい	
6098	OF1B	1708	XML PGMC	H	Advance	e point	er. Skip ",	I to The Control of t	
609A	4015	1709	BR GCHR	2	Repeat	the pro	ocessi		
		1710						TEMPLACE	ايو لايلان د . سال

```
1712 *****
                        ... 1713 ... *******
                         *****************
              1717 * ERROR MESSAGE EQUATES (msg are in console GROM)
            202C 1719 MSG1 EQU >202C Incorrect statement
2040 1720 MSG4 EQU >2040 Bad name
2049 1721 MSG5 EQU >2049 hemory full
2055 1722 MSG6 EQU >2055 Can't continue
           2049 1721 Noos
2055 1722 MS64
1723 MS63
60A6 6DAB 668C4 1747 CALL CLSNO Close the file
   1748 ERRORB
6DAB C62221 1749 $IF @ERRCOD .H. >21 GOTO ERRUEC Unknown error
6DAE 6ESC
                                                                                   6080 A62208 1750 SUB 8,8ERRCOD
                                                                         6083 8A22 4751 CASE BERRCOD
   60B5_40E9 1752 BR ERRMM 8 Memory full

      6085 40E9
      1752
      BR ERRMM
      8 Memory full

      6087 4E07
      1753
      BR ERRIS
      9 Control char -- not issued

      6089 40EE
      1754
      BR ERRTAG
      A Bad tag

      60BB 40F3
      1755
      BR ERRCHK
      B Checksum error

      60BD 40F8
      1756
      BR ERRDD
      C Duplicate definition

      60BF 40F0
      1757
      BR ERRUR
      D Unresolved reierence

      60C1 4E07
      1758
      BR ERRIS
      E No entry address -- not issued

      60C3 4E02
      1759
      BR ERRPN
      F Program not found

      60C5 4E07
      1760
      BR ERRIS
      10 Incorrect statement

      60C7 4E0C
      1761
      BR ERRBN
      11 Ead name

      60C9 4E11
      1762
      BR ERREV
      13 Bad value
```

10/17/83

09:15:56

GROM

GPL ASSEMBLER 79.150

Some the transfer of the state of

GPL ASSEMBLER 79.150	GROM	10/17/83	09:15:56	PAGE 47
	The second secon			
6DCD 4E18 41764 BR			r too bio	
60CF 4E20 1765 BR	ERRONN		g-number mism raument	s com
6001 4E25 1766 BR 6003 4E2A 1767 BR	EDDES :		ubscript	
6005 4E2F 1768 8R			conflict	
6DD7 4E34 1769 BR	ERRODT		do that	
6009 4E39 1770 BR	ERRBLN	i i	ine number	
- LOOP LETE - 4774	EBBENE	18 For-n	ext error	
6000 4E68 1772 BR 600F 4E43 1773 BR 60E1 4E48 1774 BR	ERR10B	10 1/0 e	The state of the s	A STATE OF A
600F 4E43 1773 BR	ERRFE		error	
360E1:4E48 34 1774 BR	ERRINP	1E input		
	ERRDAT		error	
6085 4852 1776 BR 6087 4089 1777 88	ERRLTL ERRMA		tco long y full withou	<b>+</b>
1778 *	ERMIN		being open	
	*********			• •
·	OR HANDLING SECT		*	v v*
	*****		***	
1782 ****				
60E9 06001C 1783 ERRMM		* Memory fu	11	
6DEC 2049 1784	DATA #MSGS			
1785 *				•
60EE 06001C 1786 ERRTAG		* Bad tag	and the second	
60F1 6E6D * ,1787 *	UNIA HUZUINU	•		
	CALL ERR\$\$	* Checksum	error	
6DF6 6E79 1790	DATA #MSGCHK	14 1 4 min 14 mi		
1791 *	ang mengangan diakan diaka	***************************************		
% 60F8%06001C% 1792€ ERRDD			definition	
60F8 6E88 4 1793	DATA #MSGDD 🤲			
1794 * 6DFD 06001C 1795 ERRUR		The second second	4+12+12 (1) + 1 + 1	
		* Unresolve	d references	
6E00 6E90 1796. 1797 *	DAIA WINSOUR	Appendix of the property		
6E02 06001C 1798 ERRPN		* PROGRAM N	OT FOUND	
6E05 6E82 5 1799		in a land		
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		4 - 4		
6E07 060010 1801 ERRIS	CALL ERRSS			
6E0A 202C 1802	DATA #MSG1	* Incorrect	statement	
1803 →				
6EUC 06001C 1804 ERRBN		m = m = 11 = 11 = 11		
6EOF 2040 1805 * 1806 *	DATA #NSG4	* BAD NAME		
4E11 06001C 1807 ERRCC			and the state of the state of	
6E14 2055 1808	DATA EMSGA	* * Canft cor	stinue.	
4 7 G G 🐷				La grand of Park Note State
6E16 06001C 1810 ERRBV	CALL ERR\$\$	and the second second		The state of the s
6E19 2064 1811	DATA #MSG7	. * BAD VALUE		
1812 *	•			
6E1B 06001C 1813 ERRNTE				
6E1E 206E	DATA #MS68	* NUMBER TO	OO BIG	
	CALL COOPE			
6E20 060010 1816 ERRSNI 6E23 2070 1817	1 CALL ERRSS	2 Charma 55	a way bay you you way on you are not be seen to	
1818 *	DATA #MSG11	* perind W	amost wramerco.	
at her an year .				

\$3.7)

0 0

```
GPL ASSEMBLER 79.150
                                                   10/17/83
                                                                  09:15:56
                                                                                   PA6E 48
 6E25 06001C 1819 ERRBA CALL ERR$$
 6E28 2094 1820 DATA #MSG12
1821 *
6E2A 06001C 1822 ERRES CALL ERR$$
                                                 * BAD ARGUMENT
                        DATA #MSG13 * BAD SUBSCRIPT
               1823
 6E20 20A1
               1824 *
 6E2F 06001C 1825 ERRNC CALL ERR$$
 6E32 20AF 1826
                              DATA #MS615 * NAME CONFLICT
1627 *
6E34 06001C 1828 ERRCDT CALL ERR$$
6E37 2080 1829 DATA #MSG17
1830 *
6E39 06001C 1831 ERRBLN CALL ERR$$
6E3C 2009 1832 DATA #MSGBLN
                                                   * CAN'T DO THAT
1833 *
6E3E 06001C 1834 ERRFNE CALL ERR$$
6E41 2DF9 1835 DATA
                              DATA #MSGBLN
                                                   * Bad line number
1836 *
6E43 06001C 1837 ERRFE CALL ERR$$
6E46 2110 ... - 1838 DATA #MEGOO
                              DATA #MSG18
                                                   * For-next error
                               DATA #MSG22
                                                   * File error
               1839 *
 6E48 05001C 1840 ERRINP CALL ERR$$
6E48 2128 1841 DATA #MSG23
                        DATA #MSG23
                                                   * Input ecror
 1842 *
6E40 06001C 1843 ERRDAT CALL ERR$$
 6E4D 06001C 1843 ERRDAT CALL ERR$$
6E50 2134 1844 DATA #MS624 * Data error
1845 *
6E52 06001C 1846 ERRLTL CALL ERR$$
6E55 213F 1847 DATA #MS619 * Line too long
1848 *
6E57 06001C 1849 ERRRMB CALL ERR$$
 6E5A 6EC4 1850 DATA #MSGRAM * No memory expansion
 1851 *
6E5C 06001C 1852 ERRUEC CALL ERR$$
6E5F 6E08 1853 DATA #NSGUEC
                                             DATA #NSGUEC * Unknown error code
 1854 *
1855 ******
 DST &SREF, &TEMP4 Copy PAB address
DSUB 04. ATEMP4 BASIC assumes offset by 4
 1862 *
6E68 06001C 1863 ERRIOB CALL ERR$$
6E6B 2113 1864 DATA #MSCCC
 6E64 A70400 1861
                        DATA #MSG21 * 1/0 error
                                                                   and the state of
                1867 ******
                      **** Error messages
                1868
                1869
               1870
                        BASE 0,0,0,0,0,0,>60
 6E6D OBA9AC 1872 MSGTAG DATA 11.: ILLEGAL TAG: 6E79 OEA3A8 1873 MSGCHK DATA 14.: CHECKSUM ERROR:
```

GPL ASSEMBLER 79.150 GROM 10/17/83 09:15:56 PAGE 49
6E88 14A4B5 1874 MSGDD DATA 20.:DUPLICATE DEFINITION:
6E9D 14B5AE 1875 MSGUR DATA 20.:UNRESOLVED REFERENCE:
6EB2 11B0B2 1876 MSGPN DATA 17.:PROGRAM NOT FOUND:
6EC4 13AEAF 1877 MSGRAM DATA 19.:NO MEMORY EXPANSION:
6ED8 12B5AE 1878 MSGUEC DATA 18.:UNKNOWN ERROR CODE:
7000 1880 ALCODE EQU >7000
1881 \* END



SDSMAC 3.4.0 81.117 13:04:56 THURSDAY, SEP 02, 1982, ACCESS NAMES TABLE

PAGE 0001

SOURCE ACCESS NAME=
OBJECT ACCESS NAME=
LISTING ACCESS NAME=
ERROR ACCESS NAME=
OPTIONS=

PCD2.ASSM.SRC.BSCSUP2
PCD2.ASSM.OBJ.BSCSUP2
PCD2.ASSM.LST.BSCSUP2
PCD2.ATA.LST.ERRS
XREF.BUNLST,TUNLST.DUNLST

MACRO LIBRARY PATHNAME=

```
13:04:56 THURSDAY, SEP 02, 1982.
                 SDSNAC 2.4.8 81.117
                                                                                     PAGE 0002
  0001
                   LDT '8SCSUP'
    0003
                               BASIC SUPPORT ROUTINES: contain numeric reference and
    0004
                            ASSIGNMENT (NUMBER, NUMBER). (STREEF, STRASG), and error reporting (ERR).
                                assignment (NUMREF, NUMASG), string reference and assig
   0006
0007
                 ODCE SET EQU >DOCE Works with console, not debugger cons
                0005
                               SYSTEM EQUATES
    0009 .
    0010
              000E SCNKEY E0U >000E
6010 EXBASX E0U >6010
200A FARM E0U >200A
8300 PAD E0U >8300
    .0011
  0012
                                                     EXTERDED BASIC XML TABLES (BASE)
                                                      information for parameter type.
     0013
     f#114 4
             8300 BYTE EQU >6300
8310 OLDS EQU >8310
  0015
    0015
                                                      ULU VALUE PTR BEFORE EVALUATE SUB
ARGUMENT
    an17 -
                                                     NO. OF ARGUMENT
                                                     PLACE TO RETURN ERROR CODE
                                                     Flag used in GPL main program.
                                                     VALUE STACK PTR
0032 83CE STFLGS EQU >83CE
0033 8300 CRULST EQU >8300
0034 8350 CRULST EQU >8300
                                                      GPL/EXTENDED BASIC WORKSPACE
0033 8300 CRULST EQU >8300
0034 83E0 GPLWS EQU >83E0
0035 *
0036 4000 WRVDP EQU >4000
  0035 *
0036 4000 WRVDP EQU >4000 Write enable for VDP
0037 8800 VDPRD EQU >8800 VDP read data address
0038 8C00 VDPWD EQU >8C00 VDP write data address
0039 8C02 VDPWA EQU >8C02 VDP write address address
  0040
 00417
                                                        Number too big
String number mism
Bad argument
Bad subscript
             1700 ERRBS EQU >1700
1800 ERRNC EQU >1800
1900 ERRCDT EQU >1900
                                                           Bad subscript
Name conflict
Can't do that
     0051
     0052
     0053
                                                           Bad line number
                1A00 ERRBLM EQU >1A00
1800 ERRFNE EQU >1800
1600 ERRIOB EQU >1600
     0854
                                                            For next error
     0055
     0055
                                                            1/0 arror
                1000 ERRFE EQU >1000
     0057
                                                            File error
     0053 1E00 ERRINP EQU >1E00
0059 1F00 ERRDAT E0U >1F00
0060 2000 ERRLTL EQU >2003
     0053
                                                            Input error
Data error
     0059
                                                            Line too lono
```

and the control of th

<u>ب</u>ر

```
SUSMAC 3.4.0 81.117 13:04:56 THURSDAY, SEP 02, 1982.
RSCSHP
                                                                        PAGE 0003
 0061
       2200 ERRSYN EQU >2200
2300 ERRNO EQU >2300
2400 ERRUC EQU >2400
                                                 Syntax error
 0062
                                                  Numeric overflow
 0063
                                                  Unrecognized character
            OFOO ERRSNE EOU >OFOO
 0064
                                                  Program not found
           2500 ERRST EQU
 0065
                              >2500
                                                 String truncated
 fiffaa.
 0067
 0058
                         UTILITY BLWP VECTORS
 0069
 0070
                          REF GPLLNK.UTLTAB
 0071
                         DEF NUMASG. NUMBEF. STRASG. STREEF. ERR
 0072 0000 0014' NUMASG DATA UTILWS NAENTR
                                                  Numerio Assignment
 0073 0004 C014' NUMBER DATA UTILWS NEEDTR
                                                  Numeric reference
 0074-0008 0014' STRASG DATA UTILWS.SAENTR
                                                  String assignment
 0075 0000 0014' STRREF DATA UTILWS.SRENTR
0076 0010 0014' ERR DATA UTILWS.ERENTR
                                                  String reference
                                                  Return error code to BASIC
 0077
 0078
                         UTILITY WORKSPACE
 0079
 0080 0014 0000 UTILKS DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           0019' R2LB EQU UTILWS+5
 0081
            0010' R4LB
 0082
                         EQU
                                UTILWS+9
 8800
                  ** DATA CONSTANTS
 0084
 0085
 0086 0034 65 CBH65 BYTE >65
0087 0035 20 EQBIT BYTE >20
              pok i 2de 🍎
```

E)

and the second second

```
BSCSUP SDSMAC 3.4.0 81.117 13:04:56 THURSDAY, SEP 02, 1982.
                                                                                                                                       PAGE 0004
         0090
                                                     NUMERIC ASSIGNMENT UTILITY
        ...0092
          0093
                                                   UTILITY ROUTINE NAENTR

A utility to allow a numeric value to be assigned
       0094
        0095
         0094
                                                        to a numeric variable or a numeric array element
       0097
                                                      passed in a parameter list of LINK call.

The numeric value to be assigned should be in FAC a
         ....0098
        0099
                                                          The stack has into. for each argument.
       0100
                                                For a simple numeric variale, the stack entry looks like this:
          0191
                                                                     IPtr to | 1>001 | IPtr to | 1
IS.T. entry | | 1 | IValue space | |
        0103
          0104
                                                                     0105
                                                            For an array element, the stack entry looks like:
           8488
           0107
                                                                    AN AND THE RESIDENCE OF THE PERSON OF THE PE
                                                                    | Ptr to | | | Ptr to | |
          0408
                                                                   1 S.T. entry | | | | Dim. info. |
          0109
        0110
                                                        REGISTER USAGE
          0112
                                                            RO : Arrav element no. or zero
        0113
0114
                                                            - R1 : Argument number
                                                           RO.R1,R2,R3,R4,R5,R6,R7,R8 are used in this
         0115
                                                       .. coutine
* 5 types of arguments:
     0127 0038 C060 MOV 202 (R13) R1 For the change of the work space
   003A 0002
0128 003C 06A0
                                                                                        Check valid argument no. and
                                                                    @CKAR6
        ****** Load parameter information address > 200A *********
                                        *****Check SIMPLE NUMERIC VAR. OR AN ARRAY REF. *********
       ... 0131. ... j. E. J.
      0132 0040 0003
                                                . MOVB *R3.R3. A Get the one byte identifier in ....
                                           *

JEQ NUERR2 Can't do the assignment to a numeric expression argument

*
      0133
0134 0042 1368 -
0135 0042
                                                                                              CPU
         0136 0044 0983
                                                             SRL
                                                                          R3.8
                                                                                           Make it double
                                                                                     Flag 2 is for NUMERIC VARIABLE
     0137 0046 0643
                                                              DECT
                                                                         RЗ
      ....0138.0048 1617 ..
                                                              JNE
                                                                         MIDMA4
                                                                                              No - co to the code for array
                                          ***** FOR ASSIGNENT TO A SIMPLE NUMERIC VAR. ********
           0139
                                                                       RO.RO In this case, RO must contain O
NUERR2 ERROR : BAD ARGUMENT
                                                       MOV
JRE
            8140 004A 0000
         - 0141 004C 1663
            0142 004E COC5
                                                            von
                                                                         RS.R3
                                                                                           Get the ptr to the stack entry
            0143 0050 0503
                                                             INGT R3
                                                                                              Point to the ID byte of the
            9144
                                                                                               stack entry
                                                                                             Set 1 byte (10) from VOP
            0145 0052 C5AC
                                                             BL
                                                                       asetv1
                     0054 03501
             2146
                                                                                              Numeric 10 byte must be 0
```

E. .

```
SDSMAC 3.4.0 81.117 13:04:56 THURSDAY, SEP 02, 1982.
      RECEILE
                                                                   . PAGE 0005
     0147 0056 165A JNE NUERRI ERROR:STRING-NO. MISMATCH | 0148 0058 0503 INCT R3 Point to the value space ptr | 0149 005A 06AD | DL | 06ET1 | Get value address |
        0050 0330
     0150 005E COC1 MOV R1.R3 Just copy
0151 0060 C103 NUN00 NOV R3.R4 PUT routine uses R4
0152 0062 0203 LI R3.FAC Load FAC address
     0264 834A
0153 0066 0202 NUMO1 LI
                                    R2.4
                                              8 bytes to copy
        0068 0004
     0154 006A 0073 NUM02 MOV *R3+.R1 Get data from FAC area
0155 006C 06A0 BL 3PUT1 Move two bytes at a time.
006E 034C'
   097A 00A24
     0163
                                            and find the otr to element in ERAM
                            JMF NUMBO Go to the general code to move from FAC to ERAM
     - 0164 0070 10F1
0165
0166
   009E 200A
0184 00A0 045B
                               RT
  0186
                    ******** CHECK THE ARRAY ELEMENT NO. 15 WITHIN BOUND *****
      0187 00A2 C24B ARYBND MOV R11.R9 Save rtn addr.
0188 00A4 0643 DECT R3 Flag 4 is for 6
0189 00A6 1632 JNE NUERR1 NO-Assume strir
                                               Flag 4 is for ARRAY NO-Assume string arg.
       0190
                                               Error:STRING-NO MISMATCH
                               MOV R5,R3
BL aGET1
                                                Get the stack entry ptr
       0191 00A8 CDC5
       0192 00AA 06A0
                                     aGET1
                                               Get the S.T. entry ptr
            00AC 0330'
       0193 00AE COC1
                               MOV R1,83
                                                R1 : Data reg. for GET1
       0194
                                                R3 : Addr. req. for GET1
                               BL aGETV1
       0195 0080 05A0
                                                Get the 1st byte of S.T. entry
           0082 03601
```

```
SDSMAC 3.4.0 81.117
                                                          13:04:56 THURSDAY, SEP 02, 1982.
                                                                                                           PAGE 0006
  0196 0084 1128 JLT NUERR1 String bit is set - ERROR 0197 *
          0198 0006 06A0
                                                  BL GARY1
                                                                           Go through this general routine!
                  0088 0002
      0199
0200 008A 6004
                                 *** R3 has address of the value space *****
                                   S R4.RO Check the BASE to 0.K.
      BASE : 1 : offset 1
                                                SLA RO.3 & bytes per element | A RO.R3 Add to the address of value space
     0202 008C 0A30
0203 008E A0C0
                                                A .
          0204 0000 0459
                                                          * 없다
      *******************************
                                                                          R1 still contains 1st byte from
          0208
       0209
0210 0006 0901
                                                                           func bits
                                                 SRL R1.13
                                                                           Shift back to use as a double
                                  SRL K1.13 Shiit dack to use as a double ****** R1:NO. OF DIMENSION, R5: STACK ENTRY PTR *******
        0211
          0212 00C8 C201
                                   NOV R1.R6 Save no. of dimension for later MOVB &BASE.R4 Check the BASE
       0213 00CA 0120
                 00CC 8343
     0214 000E 0984
0215 0000 1303
                                              SRL R4.8 Make it double
JEQ NUMOS If BASE=0.INDEX=0 -- it's o.k.
DEC RO Decrement the INDEX
JLT NUERR3 BASE=1. INDEX=0--ERROR
BAD SUBSCRIPT
INC RO Restore INDEX
    0215 0000 1303
0216 0002 0600
    0217 0004 1123
0218 *
                                                 BASE=1, INDEX=0
BAD SUBSCRIPT
INC RO RESERVE
    0219 0006 0580 INC RO RESTORE INDEX
0220 0008 0206 NUMO8 LI R6.1 Initial value for accumulator
000A 0004
0221 000C COCS MOV R5.R3 Put the stack entry ptr in R3
0220 0008 0206 NUMO8 L1
000A 0001
0221 000C 0005 MOV R5,R3 Put the stack entry ptr in R3
0222 000E 0223 AI R3.4 Try to get the dim. info. ptr
00E0 0004
0223 00E2 06A0 BL &GET1 Go to get the dim. info. ptr
00E4 0330'
0224 00E6 00C1 MOV R1,R3 Put it in R3
0225 00E8 0643 DECT R3 For the following INCT instruct
0226 00EA 05C3 NUM10 INCT R3 Point to dimension infor. entry
0227 December 2027
0228 00EC 06A0 BL &GET1 Get the dimension maxima in R1
0228 00EC 06A0 BL &GET1 Get the dimension maxima in R1
  0229 00F0 0581
                                               INC R1 BASE=0, add 1 offset

S R4,R1 BASE=1. O.K.

MPY R1,R6 Get the max.array.index

R6: accumulator.

MOV R6,R6 First 2 bytes must be 0 here

JNE NUERR3 else BAD SUBSCRIPT

MOV R7,R6 Get last two bytes result

DEC R8 Decrement the counter
 0230 00F2 6044
0231 00F4 3981
0232
                                             DEC
MOV
JWE
MOV
    0233 00F6 C186
      0234 00F8 1611
0235 00FA C187
  0236 00FC 0608
    0237
0238 00FE 15F5
0239 0100 0606
                                                                          R8 : no. of dimension
                                               JGT
DEC
                                                         NUM10 Loop for more R6 BASE=0:element
         0239 0100 0606
                                                                          BASE=0:element no. must range
  0240
0241 0102 A184
0242 0104 8180
                                                                      from 0 to (R6-1)
BASE=1:element no. range 1 -- R6
                                              A R4,R6 BASE=1:element no. range :
C R0.R6 User's INDEX must <= max. INDEX
JGT NUERR3 NO - ERROR:BAD SUBSCRIPT
LNCT R3 Contains value space address in
     - 30243 0106 150A (
         0244 0108 0503
          0245
          0246 010A 045A
                                                                           RTH
          0247
          0248 0100 0280
                                  NUERR1 LI RO.ERRENM STRING-NO. MISMATCH
                 010E 1500
```

-	Bacsup	S05MAC 3.4.0 81	.117	13:9+:56 Th	HURSDAY, SEP 02.	1982.	PAGE	0007
`. :	0249 0110 0112		3	SERROR	Error out			
		0200 NUERR2	LI	RO.ERRBA	BAD ARGUMENT			
	0251 0118 011A	10 miles (10 mil	8	@ERROR	Error out			
	0252 011 <b>C</b> 011E	0200 (NUERR3	LI	RO.ERRES	BAD SUBSCRIPT			
	0253 0120 0122	0460 039E'	. 8	. &ERROR				

(8)

```
SDSMAC 3.4.0 81.117 13:04:56 THURSDAY, SEP 02, 1982.
  BSCSHP.
                                                                          PAGE DOOR
  0255
                         NUMERIC REFERENCE UTILITY
  0256
                    *____
  0257
  0258
  0259
                          . UTILITY ROUTINE NRENTR
   0250
                         A utility to allow reference to a numeric exp.
   0261
   0282
                             passed in a parameter list of a LINK call.
   0263
                              The numeric value will be placed in the FAC
   0264
                              area after calling NRENTR.
                            The stack has info. for each argument.
   0265
                          For a simple numeric variale, the stack entry looks like this :
   0266
  0267
   0268
                                    IPtr to | 1>001 | IPtr to | 1
IS.T. entry | | 1 | IValue space |
   0269
   9270
   0271
   0272
                               For an array reference, the stack entry looks
  0273
                               like this :
   0274
                                  0275
   0276
   0277
                             for a numeric exp., the stack entry has the
  0278
                          8 bytes floating points value passed by a parse routine.
 0279
  0280
  0281
                          REGISTER USAGE
                 * RO: Array element no. or O

* R1: Argument number

* RO.R1,R2,R3,R4,R5,R6.R7,R8 are used in this

* routine
  . 0282.
 0283
0284
0285
0285 * routine
0286 *
0287 0124 C010 NRENTR MOV *R13,R0 For the change of the work space
0288 0126 C060 MOV 02(R13),R1 For the change of the work space
0128 0002
0289 012A 06A0 BL 0CKARG Check valid argument no.
                                and set up stack ptr and parm add
0290
                    ******* R5: Ptr. to the stack entry, R1: Argument no.

******* CHECK THE ARGUMENT IDENTIFIER IN CPU RAM ********
 0291
   0292
0293
                    **** A NUMERIC EXP., SIMPLE NUMERIC VAR. OR AN ARRAY *****
                     MOVB *R3.R3 Get the one byte identifier in CPU
  0294 012E 0003
0295
                              SRL R3,8 REFOO
  0296 0130 0983
                                             Kake it double
   0297 0132 1608
                                                 Not a numeric expression
                    ***** A NUMERIC EXP., MOVE THE 8 BYTES STACK ENTRY ******
   0298
                    ***** FROM VOP TO FAC.
   0299
   0300 0134 0000
                               NOV RO.RO
                                                  RO in this case must contain O
                                               ERROR : BAD ARGUMENT
   0301 0136 1629
                               JNE RFERR2
   0302 0138 0202
                               LI
                                     R2.8
                                                 8 bytes to move
        013A 00C8
   0303 0130 0204
                                     R4.FAC
                               LI
                                                 Destination addr.
        013E 834A
   0304 0140 COC5
                               MOV
                                    R5, R3
                                                 Source addr.: stack entry ptr
   0305 0142 06A0 REF03
                               BL
                                      aGET1
                                                 Get 2 bytes from VDP
        0144 03301
                               MOV R1,*R4+
INCT R3
   0306 0146 CD01
                               MOV
                                                 Put 2 bytes in FAC
   G307 0148 05C3
                                                  Update source addr.
                               DECT R2
   0308 014A 0642
                                                 Update the counter
   0309 0140 15FA
                                      REFO3
                                JGT
                                                 Nore to move
```

Ø.

0188 0100' 0344 018A 0460 RFERR2 B & NUERR2 BAD ARGUMENT 018C 0114'

0188 010C'

SOSMAC 3.4.0 81.117 | 13:04:56 THURSDAY, SEP 02. 1982.

```
SDSMAC 3.4.0 81.117 13:04:56 THURSDAY, SEP 02. 1982.
                                                  PAGE 0010
   0344
   0347
   0348
                        STRING ASSIGNMENT UTILITY
   0349
   0350
   0351
                        UTILITY ROUTINE SAENTR
   0352
                         A utility to allow a string to be assigned to a
                          string variable or a string array element passed
   0354
                          in a parameter list of a LINK call.
   0355
                          The stack has info. for each argument.
   0356
                          For a string varials, the stack entry looks like
   0.35.7
                               | Ptr to | 265| | Pointer | String
| Value space| | | Ito String| Length
   0.358
   0359
   0340
   0361
                          For a string array, the stack entry looks like
   0342
                           0343
  0364
                                            l | | Dim info.|
                               IS. T. entry!
   0365
   6366
                          USER'S REGISTER USAGE-- following regarsters must
   0367
                           be set up prior to the SAENTR call.
   0368
                           RO : Array element no. or zero
                           R1 : Argument number
   0370
                           R2 : String address in ERAM. (the first byte
   0371,
                                of the string is the length of the string.)
    0372
                      ******************
             * THIS ROUTINE REGISTER USAGE (UTILWS)

* RO : Array element

R1 - Argument number
   0373
   0374
   0375%
             R1: Argument number.
R2: Return address for VGASSN
ુ મ્રેક્ 0376 છે.
                        RS : Value stack address.
 0377
0196 0004
                                       0384.
  0385 0198 06A0 BL BL
                                 ackard Check valid argument no. and
   .0386
                                           set up ptr to stack entry in R5
  0387
                       R3 has address of parm type info.
   0388 0190 0003
                         MOVB *R3,R3
                                        Get the one byte identifier in
   0389
                                           CPU
    0390 019E 0983
                           SRL
                                 83.8
                                          Make it double
   0391 01A0 0603
                           DEC
                                 R3
                                          Check if string expression
    0392 01A2 1324
                           JEQ
                                 STERR2
                                          Can't do the assignment to a
                                        string expression argument
    0393
    0394 01A4 0643
                                          Flag 3 is for string variable NO : Is STRING ARRAY ?
                           DECT RE
    0395 01A6 163A
                           JNE
                                STARY
    0396
                        ASSIGNMENT TO STRING VARIABLE ******
    0397 01A8 C000
                          MOV
                                 RO.RO
                                         Array element has to be zero.
    0398 01AA 1620
                           JNE
                                          ERROR : BAD ARGUMENT
                                 STERR2
    0399 01AC COCS
                           MOV
                                 R5,R3
                                          Addr.req. for GETV1
    0400 01AE 05C3
                           INCT R3
                                          Point to the ID byte of the
    0401
                                          stack entry
    0402 0180 06A0
                                @GETV1
                                         Get 1 byte (ID) from VDP
```

4.

```
SDSMAC 3.4.0 81.117
                          13:04:56 THURSDAY, SEP 02, 1982.
                                                      _PAGE 0011
                                                       0182 036C
                       CB
                           R1, aCBH65 Is this a string variable ?
   0403 0184 9801
       0185 00341
                   JNE STERR1
   0404 0188 1615
                                    No - STRING NO. MISMATCH ERROR
                ***********************
   0405
                ****** PUSH THIS STACK ENTRY ON TOP OF THE STACK: FOR ******
   0406
                ***** ROUTINE "ASSGNV" TO USE
   0407%
                *******************
   0408
                ***
   0409
                     MOVE THIS STRING VARIABLE ENTRY TO CPU FAC AREA ***
                      IN ORDER TO DO A VPUSH
   0410
                ****
   0411 018A 0206
                             R6.8
                                     # of bytes to move
                       Li
   018C 0008
0412 018E 0204
                       1 1
                             R4.FAC
                                    Destination addr. in CPU
      0100 834A
   0413 0102 0005
                                     Restore stack entry ptr
                       MOV
                             R5.R3
   0414
                                     R3 : addr. req. for GET1
                                     Get 2 bytes from VDP
   0415 01C4 06A0 STR02
                       BL
                             aGET1
       0106 03301
   0416 01C8 CD01
                        MOV
                            R1.*R4+
                                     Put to FAC area
                        INCT R3
  0417 01CA 05C3
                                     Undate source addr.
   0418 01CC 0646
                        DECT R6
                                     Update the counter
   0419 01CE 15FA
                        JGT
                             STRO2
                                     Nore to move
                      DO THE VPUSH, GETSTR, COPY STRING TO STRING SPACE
   0420
   0421
                      AND ASSIGNMENT
   0422 0100 06A0 1
                      BL
                             @VGASSN
       0102 0246
 0428 0104 C105 MOV R5.R4
0429 0106 0224 AI R4,6 "STRING LENGTH" entry
0108 0006
0430 010A C060 MOV &BYTE,R1 Put the length into data reg.
010C 830C
                                     Make it two bytes
                        SRL R1,8
  0431
   0432 01DE-06A0 .
                        BL aPUT1
                                     Put 2 bytes of data to VDP
       01E0 034C'
   0433 01E2
                                             0434 01E2 0380
                        RTUP
                                     Return
   0435
                0436 01E4 0200 STERR1
                       LI RO, ERRSNM STRING-NO. MISMATCH
      01E6 1500
   0437 01E8 0460
                        В
                            @ERROR
       01EA 039E1
   0438 01EC 0200 STERR2 LI
                            ROLERRBA BAD ARGUMENT
   01EE 1600
0439 01F0 0460
                             SERROR
                                     Error return
   0440
                ******** Common routine for STRASG & STRREF to check
   0441
                ***** element
   0442
   0443 01F4 CO8B STARYC MOV R11.R2 Save RTN addr.
   0444 01F6 0643 .
                        DECT
                            83
                                     Flag 5 is for string array
   0445 01F8 16F5
                        JNE STERR1
                                     ERROR : STRING NO. MISMATCH
                             RS.R3
    0446 01FA COC5
                        MOU
                                     Get the stack entry ptr
    0447 01FC 05A0
                             aGET1
                        BL
                                     Get the S.T. entry ptr
      01FE 0330'
                             R1,R3
   0448 0200 0001
                        1104
                                    R1 : Data reg. for GET1
```

```
S0SMAC 3.4.0 81.117
                                           13:04:56 THURSDAY, SEP 02, 1982.
         0449
          R3 : Addr. req. for GET1
Get the 1st byte of S.T. entry
               0204 036C'
                                      JLT STARY1 O.K.
B SSTERR1 ERROR: STRING-NO MISMATCH
          0451 0206 1102
          0452 0208 0460
               020A 01E4
          STARY1
                           * RO has array element number
          0454
                              BL BARY1 __ Go through general routine to
          0455 020C 06A0
               020E 00C21
          0456
                                            check array element is out of bound
                            * R4:BASE, RD:ARRAY ELEMENT, R3:PTR TO BEGINNING OF VALUE
          0457
          0458
          0459 0210 6004
                                             R4.R0
                                                       Check the BASE : 0 : 0.K.
          0460
                                                                    BASE: 1: Offset 1
          0461 0212 0A10
                                            RO.1
                                     SLA
                                                         2 bytes per element
          0462 0214 AUCU
                                       A
                                              RQ.R3
                                                         Ptr to the exect element we want
          0463 0216 06A0
                                             AGET1
                                                         Get the string ptr
                                       F. 1
               0218 03301
          0464 021A 0452 .
                                       8
                                             *82
                                                         RTN
          0465
                            0466
                                     BL OSTARYC Go through general routine
          0467 021C 06A0 STARY
               021E 01F4
                                                       shared by STRREF
       0468
          0469 * NOW R1 : PTR TO THE OLD STRING
0470 * R3 : PTR TO THE VALUE SPACE
0471 *** SET UP FAC ENTRY AND PUSH IT ON STACK FOR ASSIGN ROUT.
0472 0220 0206 LI R6,FAC Destination addr.
         # 0471
        0472 0220 0206
     U4/2 U220 U206 LI R6,FAC Destination addr.

Q222.834A

0473 0224 C083 MOV R3,*R6+ Put ptr to value space in D474 U226 DDAO MOVB ACBH65.*R6+ Put >65 ID in U475 U228 U034/

0475 U22A DD84 MOVB R4,*R6+ Clear the unuse byte U475 U22C C081 MOV R1,*R6+ Put the ptr to string in U477 U22E C0C1 MOV R1,R3 Set up addr. reg. for GETV1 U478 U23U 1602 JNE STARY4 Not null ptr: o.k.

0479 U232 U406 CLR *R6 Clear the length bytes
          0479 0232 0406 CLR *R6 Clear the length bytes
         0479 0232 0000
0480 0234 1005 JMP
0481 0236 0603 STARY4 DEC
0482 0238 0640 BL
                                                     Go on
Point to string length
                                             STARY6
                                             R3
                                            aGETV1 Get the length from VDP
                                  SRL R1.8 Make it two bytes
MOV R1.*R6 Put the length in FAC area
         023A 036C
       2 0483 023C 0981
          0484 023E C581
                           ****** FAC ENTRY HAS BEEN SET UP NOW ***********
          0485
0486 0240
                           STARYS
          0487
          0488 0240 06A0
                                       eL
                                            @VGASSN
                                                        Do the VPUSH, GETSTR AND ASSIGN
0242 02461
                                    ASSIGNMENT WILL POP THE STACK, OLD STACK ENTRY ****
          0490
                                    DOES NOT NEED TO BE UPDATED IN THIS CASE ... *****
          0491 0244 0380
                                                          Return. . .
          0492
           0493
          11494
                            ***********************
          0495 0246 C088_
                            V6ASSN MOV R11.R2 Save rtn addr.
          0496
                            ******* VPUSH routine in console RON *************
           0497
                            0478 0248 0260
                                       LWP1 GPLWS Load GFL workspace.
               024A 83E0
```

Ka.

```
PECCHIP
                  SUSMAC 3.4.0 81.117 13:04:56 THURSDAY, SEP 02, 1982.
                                                                                                                    PAGE 0013
                                            LI R11.>1EAA XML table addr. for VPUSH
 0499 024C 020B
          024E 1EAA
  0500 0250 069B
                                                          *R11
                                              8L
                                                                             Do the VPUSH
                                             LWPI UTILWS
  0501 0252 02E0
                                                                             Reload utility workspace.
         0254 00144
  0502
                             0503
                          ***** GETS THE STRING SPACE. COPIES THE STRING FROM ERAM **
                             ***** INTO VDP STRING SPACE, AND SETS UP THE FAC WITH A **
***** STRING ENTRY OF THE FOLLOWING FORM: **
  0504
  0505
                                        TOTAL CONTROL 
                             ****
                             ***** | >001C | >65 | XX | POINTER | LENGTH |
  0507
  0508
                                        1 1 1 TO STRING | OF STRING |
                             ****
  0509
                             ***** FAC +2 +3 +4
  0510
  0511
                             MOVB *R9.R6 R9:Addr. of string (length byte
  0512 0256 0199
  0513
                                                         & string) supplies by ALC R6.8 Set up the length byte
                                               SRL R6.8 Set up the length byte
MOVB R6.8STATUS MAKE SURE NO ACCIDENTAL ERROR EX
  0514 0258 0985
  0515 025A D806
          0250 8370
  0516 0258
  0517 025E C806
                                               MOV
                                                          R6. REYTE For GETSTR routine
         0240 8300
  0518 0262 0806
                                               MOV
                                                          R6.2FAC6 : Set up length byte entry in FAC
         0264 8350
  0519
                              ******** GETSTR routine in console GROM ***********
  0520
  0521 0266 0420
                                            BLWP @GPLLNK | Call GETSTRG routine
         0268 0000
                              0522 026A 0038
  0523
0524 026C 0206
026E 834A
0525 0270 0204
                                                         R4,>001C Get addr. of SREF
                                          LI
                                    MOV R4,*R6+ Indicate a temporary string
  0526 0274 CD84 //
0527 0276 DDAO
0278 0034
                                             MOV8 OCBH65.*R6+ Indicate a string
                                             MOVB R4.*R6+ Byte is not used
  0528 027A 0084
  0529 027C_CSA0
                                               MOV
                                                          aSREF.*R6 Save ptr to the string
                                                          COPY THE STRING TO STRING SPACE *********
          027E 831C
  0531 0280 C220 (2) ( ) MOV
                                                          abyte, Ra Get # of bytes to copy
          0282 8300
  0532 0284 1309 -
                                                JEQ
                                                          STRO4
                                                                            If none to copy
  0533 0286 C116
                                                                          Get pointer to destination
                                               NOV
                                                          *R6.R4
  0534
                                                                            R4:addr. reg. for PUTV1 below
  0535 0288 0009
                                               MOV
                                                          R9.R3
                                                                             R9:Ptc to string (length byte)
  0536 028A 0583
                                               INC
                                                          8.3
                                                                            Skip the len byte-ptr to string
  0537 0280 0073 STR06
                                             MOVB *R3+,R1
                                                                            Get one byte from ERAM
  0538 028E 06A0
                                               BL
                                                          @PUTV1
                                                                            Put 1 byte in VDP
          0290 03804
  0539 0292 0584
                                               INC
                                                          84
                                                                             Undate the destination addr.
  0540 0294 0608
                                                DEC
                                                          8.8
                                                                             1 less to move
  0541 0296 15FA
                                                JGT
                                                          STRUS
                                                                             If not done-loop for more
  0542
                                                                            ASSGNV destroys all the reg.
  0543
                                                Call ASSNV routine in console ROM **********
                              *****
  0544
                              ****
  0545 0298 02E0 STR04
                                               LRPI GPLRS
                                                                             Load GPL worksmade
          029A 83EG
```

And the second second

0

0	BSCSUP	SUSMAC 3.4.0 81.	117 13:04:56 TH	URSDAY. SEP 02. 1982.
	0546 029C	020B 1788	LI R11.>1788	PAGE D014 Access to ASSGN directly
<b>@</b>	0547 02A0 0548 02A2 02A4	0698 02E0	BL *R11 LWPI UTILWS	Call ASSGNV Re-load utility workspace
6	0549	******	MOV R2,R11 RT B *H2	Restore RTN addr.
0				

£.

```
SOUNAU 3.4.0 81.117 15:04:56 THURSDAY, SEP 02. 1982.
   RECEIP
                                                                    PAGE 0015
    0553
    0554
                          STRING REFERENCE UTILITY
    0555
    0556
    8557
                           HILLITY ROUTINE SRENTR
    0558
                             A utility to allow a ref. to a string expression.
    0559
                             a string variable or a string array element passed
    0550
    0561
                             in a parameter list of LINK call.
    0562
                             The stack has info. for each argument.
    05.63
    0554
                             For a string expression, the stack entry looks
    0545
                              likes this
    0566
                                 ---
                                 | >001C | >65| | Pointer | String | for Ptr to | | | 1to String | Length | | | | | | |
    0567
    0568
    0569
    0570
                                 >0010 : For a temporary string
    0571
    0572
                                 Ptr to Value Space : for a permanent string
    0573
                             For a string variale, the stack entry looks like
                             $13.2 M
    0574
                                 (Ptr to | 1>65| | Pointer | String |
(Value space| | 1 | to String| Length | 1
    0575
    0574
    0577
    0578
                             MOTE: in this case, the pointer to the string may
                              no longer be valid.
    0579
                              For a string array, the stack entry looks likes
    0580
    0581
    0582
  0583
                                 0584
                                  0585
 OS8A.
~ ~ ~ 0587 ....
                            USER'S REGISTER USAGE -- the following registers mu
                              be set up prior to the call.
    0588
0589
                             RO : Array element or zero
 0590
                              R1 : Argument number
   0591
                              R2 : String address in ERAM. (the first byte
                                   of the string is the length of the string.)
    8592
   0593
                                   Before calling SRENTR : The string len has
                                   the max. length which can be allocated for . .
    0594
    0595
                                   the string.
                                   After SRENTR call: The string len is the actu
    0594
                                   length of the string which is followed by the
    0597
    0598
                                   string itself.
    0599
                    ************
    0.600
     0401 0248 CQ10
                    SRENTR MOV *R13,RG for the change of the work space
                                              Array element now is in RO
    0.602
                                   82(R13).Ri For the change of the work space
     0603 02AA C060
                              500
         02AC 0002
                                              Argument no. is in R1
     0604
     0605 02AE 06A0
                                   <u>a</u>CKARG
                                              Chack valid argument no. and
         0280 007E4
                                              set up ptr to stack entry in RE
     0.604
                    * RS points to parm type into address
     0607
     0608 0262 0003
                     MOVB *R3.R3 Get one byte identifier in CPU
     0509 0284 0983
                              SRL
                                    R3,8
                                              Make it double
                                              Flao 1 is for string expression
     0510 0286 0603
                              DEC
                                   원품
```

DEC

13:04:56 THURSDAY, SEP 02, 1982.

SDSMAC 3,4.0 81.117

BSCSUP

0561 0310 0505

BSCSUP	SOSMA	0 3.4.0	81.117	13:04:56 Th	URSDAY, SEP 02, 1982.
				•	PAGE 0017
0662 03	12 15FA	Karalan da karangan da kar	JGT	SRFO7	More to move - loop back
0663 03	14 0380	SRF08	RTWP	25.0	Return the caller
0884		****	******	*****	**********
0665		****	**** RE	F. TO STRIM	16 ARRAY ************
0666 03	16 06A0	STARYF -	BL -	astaryc	Check array element is within
	18 01F4*				
0667	+ 1 - 41 - 1	*	bound and	I find the s	string element ptr in value space
0668 Q3	1A 0020 :		Von	_84(R13),RD	) Put provided string addr.in ERAM
03	10,0004	i garting i e			
0659		.* by t	he user i	nto RO now.	(for the change of work space)
0670-03	1E 10E6		JMP	SAF03	Go to the general code to move the string from VDP to ERAM
0671		*			the string from VDP to ERAM
0673 03	20 0460	SRFER1	В	anuerri	STRING-NO. MISMATCH
03	22 010¢′				
0574 03	24 0460	SRFER2	₿	<b>anuerr</b> 2	BAD ARGUMENT
03	26 01147	*		•	
0675 03	28 0200	SRFER3	LI	RO.ERRST	STRING TRUNCATED
03	2A 2500				
0676 03	20 0460		В	<b>@ERROR</b>	Error out

032E 039E1

(E)

```
SDSMAC 3.4.8 81.117
                                    13:04:56 THURSDAY, SEP 02, 1982.
                                                                         PAGE 0018
      0678
                      *** UTILITY ROUTINES TO ACESS VOP
      0840
                      *******************************
      0681
                      ***** GET1 : Get two bytes of data from VDP
                      ***** R3 : Address in VDF
      9682
      0683
                      ****
                                 R1 : Where the two bytes data stored
      0683
0684 0330 0603
                      GET1 SWPB R3
MOVB R3. QVDPWA
      0685 0332 D803
          0334 6002
                               SUPB
      0686 0336 0603
                              nove R3.avopua
      0687 0336 0803
           033A 8C02
      0688 033C 1000
0689 033E 0060
                                 MOP
                                MOVB &VDPRD.R1
          0340 8800
      0690 0342 0601
                                 SWPB
                                       R1
      0691 0344 0060
                                      avopro.R1
                                MOVE
          0346 8800
      0692 0348 0601
                                 SWPB R1
      0693 034A 0458
                                 RT
      0694
                      ***** PUT1 : Put two bytes of data into VDP
      0695
                      ***** R4 : Address on VDP
      0696
                      *****
                                 R1 : Data
     0697 0340 0604 PUT1
0698 034E 0804 PUT1
                                 SWPB R4
                                MOVE R4. &VOPWA
           0350 8002
     _ 0699 0352 06C4 .... ...
                              SWPB R4
  0700 0354 0264
0356 4000
                                 ORI R4, NRVDP
   0701 0358 D804
0358 8C02
                                MOVB R4, EVDPWA
0702 035A 8C02

0702 035C 1000 NOP

0703 035E 0801 NOVB R1.0VDPWD

0360 8C00

0704 0362 06C1 SWPB R1

0705 0364 0801 MOVB R1.0VDPWD

0366 8C00

0706 0368 06C1 SWPB R1

0707 0364 045B RT
                                                          0707 036A 045B
   0707
0708
                                 RT
                      ****** GETV1 : Get 1 byte of data from VDP
   0709
                      ***** R3 : Address in VDP
    0710 ******
0711 0360 0603 GETV1
                                  R1 : Where the 1 byte data stored
                                 SWPB R3
     0712 036E D803
                                 MOVB R3.2VDPWA
           0370 8002
      0713 0372 0603
                                 SWPB R3
      0714 0374 0803
                                 MOVE
                                      R3.8VDPWA
           0376 8002
      0715 0378 1000
                                 MOP
      0716 037A 0060
                                 MOVE averse.R1
           0370 8800
      0717 037E 045B
      G718
                       ***** PUTV1 : Put 1 bytes of data into VDP
      0719
                       ****
                              R4 : Address on VOP
      0720
                       经安全条件
                                  R1 : Data
      0721 0380 0604
                      PUTV1
                                 SKPB R4
      0722 0382 0804
                                 MOVE R4.8VDPMA
           0384 8002
      0723 0386 0604
                                 SUPB R4
      0724 0388 0264
                                 ORI
                                       R4.WRVDP
```

And the second of the second o

Con.	BSCSU	D.	SUSMAC	3,4.0	81.	117	13:04:56 T	HURSI	DAY. S	EP 02.	1982.	PAGE	0019
	0725		4000 0804			MOVB	R4. @VDPWA						
	0726	038E 0390	8002 1000			404	·	÷					
	0727	0394	8000			NOVE RT	R1.avopwo						
										. :			
€0\.			* . * - * a										

SDSMAC 3.4.8 81.117 13:04:56 THURSDAY, SEP 02, 1982. PAGE 0020 0730 ERROR ROUTINE \*\* 0731 0732 ERERTR MOV 0733 0398 C810 \*R13.8ERRCOD 039A 8322 0734 039C 1002 0735 039E C800 ERROR MOV RO, BERRCOD 03A0 8322 ERR2 LWPI GPLWS 0736 03A2 02E0 03A4 83E0 R11.UTLTAB 0737 03A6 020B LI 0000 8AE0 0738 03AA 022B AI R11.14 03AC 000E 0739 03AE C2DB MOV \*R11.R11 restore GPL return address 0740 0380 0460 8 SSET 0382 00CE 0741 FND NO ERRORS. NO WARNINGS

200 00 00 00 1 00												
BSCSUP	SUSMAC	3.4.0	81.11	7 13	:04:56	THURSE	PΑΥ.	SEP U2.	1982.			
LABEL	VALUE (	DEFN R	REFERE	NCES						PAGE	0021	
								124 16	e formalist of	and the point	4.00	
ARY1	00021	0206	0198	0455:		5 - 4			A 125 C	general de la		
ARYEND	00A2'		0162								a first transfer	
		0021		6336			1.		make again to the	in garagas	1.5	
BASE	8343		0213									
BYTE	8300	0015	0430	0517	0531							
CBH45	00341	.0085	0403	0474	0527	0623		·	100			
CKARG	097E' ·	0169	.0128	0289	0385	0605		15 5 5 5		4 - 42		
COUNT	8312	0018	0172						100		Section 1	
CRULST	8300	0033					1.0					
EGBIT	00351	0087										
ERENTR	0398'	0733	0076				* .	11.772	100			
ERR D	00101	0076	0071									
				•								
ERR2	03A2	0736	0734	error en encomo					•			
ERRBA	1600	0050	0250	0438					•			
ERRBLN	1 A 0 0	0054										
ERRBN	1100	0044										
ERRBS	1700	0051	0252									
ERRBV	1300	0047										
ERRCC	1200	0046										
ERRODT		0053						*				
ERRCOD	8322	8020	0733	0735								
		0059	0/35	0722								
ERRDAT	1F00											
ERRFE	1000	0057										
ERRFNE	1800	0055										
ERRINP	1E00 .	.0058										
ERRIOS	1000 3	0056						100	4 · 4			
ERRIS	1000	0043										
ERRLTL	2000	0060				_		100				
ERRMM	2100	0045		1.0		t.				e e		
ERRNC		0052			100	<i>ii</i>			7 - N - 1 - 1 - 1 - 1	4.5°		
ERRNO						14					100	
		カベム・フ	4.5		A Section 1985	and the first			, y		and the second second	
	2300	0062 (	N. W. A.		1 1 + 2≥0 2 An 1 168	ages Tigger	o the tweety	. The Marketing		2. 165	and the state of t	
ERRNTB	1400	0048		000		24.77	i i gaya			5. M. (		544
ERRNTB ERROR	1400 039E4	0048 0735	0249	0251	0253	0437	0439	7 0676				Mil
ERRNTB ERROR ERRSNF	1400 039E' OFOO	0048 0735 0064			0253	0437	<b>843</b> 9	7 0576				
ERRNTB ERROR ERRSNF ERRSNM	1400 039E' 0F00 1500	0048 0735 0064 0049	0248	0251 0436	0253	0437	0439	7 0676				
ERRNTB ERROR ERRSNF	1400 0396′ 0F00 1500 2500	0048 0735 0064			0253	0437	0439 	7 0676				
ERRNTB ERROR ERRSNF ERRSNM	1400 0396' 0F00 1500	0048 0735 0064 0049	0248		0253	0437	8439 10439	7 0676				
ERRNTB ERROR ERRSNF ERRSNM ERRST	1400 0396′ 0F00 1500 2500	0048 0735 0064 0049 0065	0248		0253	0437	0439	7 0676				
ERRNTB ERROR ERRSNF ERRSNM ERRST ERRSYN	1400 039E' 0F00 1500 2500	0048 0735 0064 0049 0065 0061 0063	0248		0253	8437	0439 9439	7 0576				
ERRNTB ERROR ERRSNF ERRSNM ERRST ERRSYN ERRUC EXBASX	1400 037E' 0F00 1500 2500 2200 2400 6010	0048 0735 0064 0049 0065 0061 0063 0012	0248 0675	0436					0524			
ERRNTB ERROR ERRSNF ERRSNM ERRST ERRSYN ERRUC EXBASX FAC	1400 037E' 0F00 1500 2500 2200 2400 6010 834A	0048 0735 0064 0049 0065 0061 0063 0012 0023	0248 0675 0027		0253	0437 0327	0439		0524			
ERRNTB ERROR ERRSNF ERRSNM ERRST ERRSYN ERRUC EXBASX FAC FAC6	1400 037E' 0F00 1500 2500 2200 2400 6010 8344 8350	0048 0735 0064 0049 0065 0061 0063 0012 0023	0248 0675	0436					0524			
ERRNTB ERROR ERRSNF ERRSNM ERRST ERRSYN ERRUC EXBASX FAC FAC6 FLAG	1400 037E' 0F00 1500 2500 2200 2400 6010 834A 8350 8348	0048 0735 0064 0065 0061 0063 0012 0023 0027	0248 0675 0027 0518	0436	0303	0327	041:	2 0472	5.5	0/45	8/47	
ERRNTB ERROR ERRSNF ERRST ERRSYN ERRUC EXBASX FACC FACC FACC GET1	1400 037E' 0F00 1500 2500 2200 2400 6010 8344 8350	0048 0735 0064 0065 0061 0063 0012 0023 0027 0022 0684	0248 0675 0027 0518 0149	0436 0152 0192	0303 0223	0327		2 0472	0524 0329	0415	0447	
ERRNTB ERROR ERRSNF ERRST ERRSYN ERRUC EXBASX FAC FAC6 FLAG GET1	1400 039E' 0F00 1500 2500 2500 2400 6010 834A 8350 8348 0330'	0048 0735 0044 0049 0065 0061 00043 0012 0023 0027 0022 0684	0248 0675 0027 0518 0149 0463	0436 0152 0192 0626	0303 0223 0632	0327	041:	2 0472 5 0324	0329		San San	
ERRNTB ERROR ERRSNF ERRSNM ERRSYN ERRUC EXBASX FAC FAC6 FLAG GET1	1400 039E' 0F00 1500 2500 2400 6010 834A 8350 8348 0330'	0048 0735 0064 0049 0065 0061 0063 0012 0023 0027 0022 0684	0248 0675 0027 0518 0149 0463 0145	0436 0152 0192	0303 0223 0632	0327	041:	2 0472 5 0324	0329	0415 0640	0447 0658	
ERRNTB ERROR ERRONF ERROYN ERROYN ERRUC EXBASX FAC FACA FLAG GET11 GETV1 GPLLNK R	1400 039E' 0F00 1500 2500 2200 2400 6010 834A 8350 8348 0330' 036C' 0268'	0048 0735 0044 0049 0065 0061 00023 00123 0027 00227 0084 0711	0248 0675 0027 0518 0149 0463 0145 0521	0436 0152 0192 0626 0195	0303 0223 0632 0320	0327	041:	2 0472 5 0324	0329		San San	
ERRNTB ERROR ERRSNF ERRSNM ERRSYN ERRUC EXBASX FAC FAC6 FLAG GET1	1400 039E' 0F00 1500 2500 2400 6010 834A 8350 8348 0330'	0048 0735 0064 0049 0065 0061 0063 0012 0023 0027 0022 0684	0248 0675 0027 0518 0149 0463 0145	0436 0152 0192 0626	0303 0223 0632 0320	0327	041:	2 0472 5 0324	0329 0622	0640	San San	
ERRNTB ERROR ERRONF ERROYN ERROYN ERRUC EXBASX FAC FACA FLAG GET11 GETV1 GPLLNK R	1400 039E' 0F00 1500 2500 2200 2400 6010 834A 8350 8348 0330' 036C' 0268'	0048 0735 0044 0049 0065 0061 00023 00123 0027 00227 0084 0711	0248 0675 0027 0518 0149 0463 0145 0521	0436 0152 0192 0626 0195	0303 0223 0632 0320	0327	041:	2 0472 5 0324	0329 0622		San San	
ERRNTB ERROR ERRONF ERROYN ERROYN ERROC EXBASX FAC FAC6 FLAG GET1 GETV1 GPLLNK R GPLUS	1400 037E' 0F00 1500 2500 2400 6010 834A 8350 8348 0330' 036C' 0268' 83E0 8370	0048 0735 0044 0049 0065 0061 00023 0012 0027 0027 0027 0084 0711 0070 0034	0248 0675 0027 0518 0149 0463 0145 0521	0436 0152 0192 0626 0195	0303 0223 0632 0320	0327	041:	2 0472 5 0324	0329 0622	0640	San San	
ERRNTB ERROR ERRSNF ERRSNN ERRSYN ERRUC EXBASX FAC6 FLAG GET1 GETV1 GPLLNK R GPLUS NAXVDP NAENTR	1400 037E' 0F00 1500 2500 2400 6010 834A 8350 8348 0330' 036C' 0268' 83E0 8370 0036'	0048 0735 0044 0049 0065 0061 0063 0012 0027 0027 00684 0711 0070 0034 0029 0126	0248 0675 0027 0518 0149 0443 0145 0521 0498	0436 0152 0192 0626 0195	0303 0223 0632 0320	0327	041:	2 0472 5 0324	0329 0622	0640	San San	
ERRNTB ERROR ERROR ERROYN ERROYN ERROC EXBASX FAC6 FLAG GET1 GETV1 GPLLNK R GPLUS MAXVDP NAENTR NRENTR	1400 037E' 0F00 1500 2500 2400 6010 834A 8350 8348 0330' 036C' 036C' 036C' 036C' 036C' 036C' 036C' 036C'	0048 0735 0064 0049 0065 0061 0063 0027 0023 0027 0024 0711 0070 0034 0029 0126 0287	0248 0675 0027 0518 0149 0463 0145 0521 0498 0072 0073	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045(	2 0472 5 0324 3 0482	0329 0622	0640	San San	
ERRNTB ERRSNF ERRSNM ERRSYN ERRSYN ERRUC EXBASX FACG FACG FALAG GET1 GPLLNK R GPLLNK R GPLLNK NAXVDP NAENTR NRENTR NUERR1	1400 037E' 0F00 1500 2500 2400 6010 834A 8350 8348 0330' 036C' 0268' 83E0 8370 0036' 0124' 010C'	0048 0735 0064 0065 0061 0063 0012 0027 0027 0027 0028 0711 0034 0029 0126 0287 0248	0248 0675 0027 0518 0149 0463 0145 0521 0498 0072 0073 0147	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0640	San San	
ERRNTB ERRSNF ERRSNM ERRSYN ERRSYN ERRUC EXBASX FAC FAC6 FLAG GET1 GPLLNK GPLLNK GPLLNK GPLLNK GPLLNK NAENTR NHENTR NUERR1 NUERR2	1400 037E' 0F00 1500 2500 2400 6018 834A 8350 8348 0330' 036C' 0268' 83E0 83F0 0124' 010C' 0114'	0048 0735 0064 0065 0061 0063 0012 0027 0027 0027 0028 0711 0034 0029 0126 0287 0248 0250	0248 0675 0027 0518 0149 0463 0145 0521 0498 0072 0073 0147 0134	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045(	2 0472 5 0324 0 0482	0329 0622	0640	San San	
ERRNTB ERRSNF ERRSNM ERRSYN ERRSYN ERRUC EXBASX FACC FACC FACC FACC FACC FACC FACC FAC	1400 037E' 0F00 1500 2500 2400 6010 834A 8350 8348 0330' 036C' 0268' 83E0 8370 0036' 0124' 010C' 0114' 011C'	0048 0735 0064 0065 0061 0063 0012 0027 0027 0027 0034 0034 0029 0126 0287 0248 0250 0252	0248 0675 0027 0518 0149 0463 0145 0521 0498 0072 0073 0147 0134 0217	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0640	San San	
ERRNTB ERRSNF ERRSNM ERRSYN ERRSYN ERRSYN ERRUC EXBASX FAC FACA FLAG GET1 GPLLNK GPLLNK GPLLNK ROPLLNS NAENTR NUERR1 NUERR2 NUERR3 NUMOO	1400 037E' 0F00 1500 2500 2400 6018 834A 8350 8348 0330' 036C' 0268' 83E0 8370 0124' 010C' 0114' 011C' 0060'	0048 0735 0064 0065 0061 0063 0012 0023 0027 0684 0711 0034 0029 0126 0250 0252 0151	0248 0675 0027 0518 0149 0463 0145 0521 0498 0072 0073 0147 0134	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0640	San San	
ERRNTB ERRSNF ERRSNM ERRSTN ERRSYN GETT1 GETT1 GETT1 GPLUS MAXVDP MAENTR MUERRS MUERRS MUMBRS MUMDO MUMO1	1400 037E 0F00 1500 2500 2500 2400 6010 834A 8350 8348 0330 0268 83E0 8370 0036 0124 010C 0114 011C 006G	0048 0735 00449 0065 00063 00063 00027 00227 00228 0711 00234 00227 02248 0225 0225 0225 0225 0225 0225 0225 022	0248 0675 0027 0518 0149 0463 0145 0521 0498 0072 0073 0147 0134 0217 0164	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0648	0658	
ERRNTB ERRSNF ERRSNM ERRSTN ERRSYN GELUS MALENTR MUERRS NUMPO NUMPO NUMPO NUMPO	1400 037E/ 0F00 1500 2500 2400 6010 834A 8350 8348 0330/ 0268/ 83E0 8370 0036/ 0124/ 010C/ 0114/ 011C/ 006G/ 006A/	0048 0735 00649 0065 00063 00027 00227 00227 00227 00227 00224 00226 0026 0	0248 0675 0027 0518 0149 0463 0145 0521 0498 0072 0073 0147 0134 0217 0164	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0640	0658	
ERRNTB ERRSNF ERRSNN ERRSTN ERRSTN ERRSYN GETUN	1400 037E/ 0F00 1500 2500 2400 2400 834A 8350 8348 0330/ 036C/ 0268/ 83F0 0036/ 0124/ 010C/ 0114/ 011C/ 0066/ 006A/ 0078/	00485 00497 00497 0065 00063 000227 00687 000227 00687 000227 00224 00225 0025 00	0248 0675 0027 0518 0149 0463 0521 0498 0072 0073 0147 01347 0164 0158	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0648	0658	
ERRNTB ERRSNF ERRSNM ERRSTN ERRSYN GELUS MALENTR MUERRS NUMPO NUMPO NUMPO NUMPO	1400 037E/ 0F00 1500 2500 2400 6010 834A 8350 8348 0330/ 0268/ 83E0 8370 0036/ 0124/ 010C/ 0114/ 011C/ 006G/ 006A/	0048 0735 00649 0065 00063 00027 00227 00227 00227 00227 00224 00226 0026 0	0248 0675 0027 0518 0149 0463 0145 0521 0498 0072 0073 0147 0134 0217 0164	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0648	0658	
ERRNTB ERRSNF ERRSNN ERRSTN ERRSTN ERRSYN GETUN	1400 037E/ 0F00 1500 2500 2400 2400 834A 8350 8348 0330/ 036C/ 0268/ 83F0 0036/ 0124/ 010C/ 0114/ 011C/ 0066/ 006A/ 0078/	00485 00497 00497 0065 00063 000227 00687 000227 00687 000227 00224 00225 0025 00	0248 0675 0027 0518 0149 0463 0521 0498 0072 0073 0147 01347 0164 0158	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0648	0658	
ERRORE ER	1400 037E/ 0F00 1500 2500 2400 2400 834A 83548 0330/ 036C/ 0268/ 83E0 8370 0124/ 010C/ 011C/ 011C/ 0066/ 0078/ 0008/ 0008/	00485 0735 0049 00061 00062 000227 000227 000227 000224 000224 000225 00153 00226 00226	0248 0675 0027 0518 0149 0443 0521 0498 0072 0673 0147 01347 0164 0128 0215 0238	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0648	0658	
ERRORE ER	1400 037E/ 0F00 1500 2500 2400 834A 8350 8348 036C/ 0268/ 83E0 8370 0124/ 0110C/ 0114/ 0116/ 0066/ 0068/ 000	00485 00735 00049 00065 00061 000227 00027 00	0248 0675 0027 0518 0149 0145 0521 0498 0072 0073 0147 0134 0217 0164 0138 0215	0436 0152 0192 0626 0195 0545	0303 0223 0632 0320 0736	0327 0228 0402	041: 030: 045:	2 0472 5 0324 0 0482	0329 0622	0648	0658	

BSCSUP	SOSMA	 6 3.4.0	81.11	7 13	:04:56	THURS	DAY. S	EP 02.	1982.		
LABEL	VALUE	DEFN	REFERE	NCES			•		and the	PAG	E 0022
P. P.O.				1.5					为政体验 主		A Are State
OLDS	8310	0016		4.1					Alta Maria		
	8300		0627	· · · · ·			1	And States			
PARM	200A	0013								2 - 1 L	
	03401	0697	0155	0432					4.505	1.	
PUTV1	0380′	0721	0538								•
RO .	0000	*	0126			0200		0203		0219	
e · · · · · · · ·			0248				0300			0315	
			0397			0438	0459		0462		0616
			0616	0518	0642	0546			0675	0735	
R1	0001		0127			0169		0171		0174	0175
			0181	0193		0210		0224		0230	0231
			0288	0304		0330		0403		0430.	
			0476			0484		0403	8523	0627	0629
			0634	0634	8535	0637	0639	0642	0646	0656	0457
			0659	0459	0690	0091	0692	0763	0704	0705	0706
			0716	0727							
	660A		0206	0246					* .		
R11 .	0008		0187	0206	0443	0495	0499	0500	0546	0547	0737
			0738	0739	0739						
R13	0000		0126		0287	0288	0381	0382	0383	0501	0403
			0618	0448	0733						
R2	0002		0153	0157	0301	0308	0328	0332	0443	0464	0475
		•	0551			* * * * * * * * * * * * * * * * * * * *					
	00194	0081									Company of the second
83	0003	•	0132	0132	0136	0137	0142 .	0143	0148	0150	0151
1.11	And the second		8152	0154	0181		0183	0188	0191	0193	0203
	د از وادید میدند و معامله معامله		0221	.0222	0224	0225	0226	0244	0294		0296
Gradiet t			0304	0307	0312	0317	0318	0323	0325	0331	0388
			0388	0307 0390	0391	0394	0399	0400		0417	
富古城的 沙沙沙		San Control	0446	0448	0442	0473	0477	11484	OE ZE	0574	ne zz 🐇 🔻 .
# Office And State		Tyle day	0608	0508	0609	0610	0612	0619	0620	0625	0629
en garage garage de la colonia	active two cars	. 4.5.4	0631	0639	0653	0660	0684	0485	0686	0687	
		1.6.0	0712	0713	8714	Dan Halis	e de la composición dela composición de la composición dela composición de la composición de la composición dela composición dela composición de la composic				
R4: "	.0004		0151	0156	0200	0213	0214			0303	0306
		100 m	0327	0330	0412	0416	0428	0429	0459	0475	0525
		-3.	0526	0528		0539		0698		6700	
			0721	0722	0723	0724	0725		e		
R4LB	0010'	_ 0082			2.4	, ,					
R5	0005		0142	0175	0176	0177	0179	0191	0221	0304	0317
			0399	0413	0428	0446	0619	0657			
R6	0006.	Spin (	0220	0231	0233	0233	0235		0241	0242	0411
100		- 1	0418	0472	0473	0474	0475	0476		0484	
a Na Marka a la	::	18	0514	0515	0517	0518	0524				0529
			0533	0636	0653						
R7	0007		0235							14.	•
R&	0008		0212	0236	0534	0540					
R9	0009		0187		0383	0512	0535			, i	* *
REFOO	01501	0312	0297				<b>3</b>				
REFCi	Q16A'	0327	0340								
REFO2	01727	0329	0333								
REF03	01427	0305	0309							*	
REFO4	01801	0338	0313		•					\$1.5	
RFERR1	01861	0343	0322								•
RFERR2	018A'	0344	0301	0316			•				
	0185	0381	0074							100	
SCLEN	8355	0025	erver I T								
SCNAME	8354	0025									
SCHKEY	DOOE	0011									
SCIEMP	3354	0024									
SET	OOCE	0007	0740								

0000

6

•

6.

(Prof.)

	85CSUP LABEL	SOSMAC VALUE - 0	3.4.0 EFN F			04:56	THURS	AY. SE	P 02. 1	982.	PAGE	0023
	SNOADD	83CC	0031	1	4 1 4 14		,			- 1 T		
	SREF	8310	0019	0529	0627							
	SREKTR	02A8	0601	0075								
	SRFO1	02BE'	0615	0611								
	SRF02	02E4	0631	0628								•
	SRF025		0632	0630								
	SRF03	02EC'	0634	0670						***	100	100
	SRFOS	O2FE'	0646	0635					•			
	SRF07	0308'	0658	0662								
	SRFC8	0314	5643	0649								
	SRFER1	0320'	0673	0624								
	SEFER2 .	03241	8674	0617								
	SRFER3	03281	0675	0544								
	STARY	021C'	0467	0395								
,	STARY1	020¢′	0453	0451								
	STARY4	0236′	0481	0478								
	STARY6	0240′	0486	0480								
	STARYC	01F41	0443	0467	8666							
	STARYF	0316′	0455	0613								
	STATUS	837C	0030	0515								
	STERR1	01E4'	0436	0404	0445	0452						
•	STERR2	O1EC,	8438	0391	0398							
	STFLGS	83CE	0032									
	STRO2	0104	0415	0+19								
	STRO4	02981	0545	0532	•				-			
	STROS	0280	0537									
i - i -	STRASG D		0074	0071						e e e e e	100	100
	STRREF D			0071						7 1 to 1		
	STRTN	- 01E2'								100		
14 de 14	UTILNS Se			0072	0073	0074	0075	0076	0081	0082	0501	0548
	UTLTAB R		0070						1 1 1 1 2			70.00
	VDPRD:	8800		0589	0691	0716			No.	all the s	ar day of	A. W.
$C_{\tau}$	VDPWA	8002	0039	0685	0687	0698	0701	0712	0714	0722	0725	
			- 0038		0705	0727	4					
	VGASSN			0422	0488	100		•				
. '		836E	0028						4.4			
100	WRVDP.	4000	0036	0700	.0724				1. 1. 1. 1. 1. 1. 1. 1.		ganishi shirin	an esta 1. D
	and the second second								12	N 96 157		

(

(\*)

(

0



\$08860 3.4.0 81.117 08:54:13 MONDAY. JAN 25, 1982. ACCESS NAMES TABLE

PAGE 0001

SOURCE ACCESS NAME=
OBJECT ACCESS NAME=
LISTING ACCESS NAME=
ERROR ACCESS NAME=
OPTIONS=
MACRO LIBRARY PATHNAME=

SD2.ASSM.SRC.SAVE2 SD2.ASSM.OBJ.SAVE2 SD2.ASSM.LST.SAVE2

XREF.BUNLST, TUNLST. DUNLST

```
PAGE 0002
                      1BT 'SAVE'
   กกกว
   0003
                      Save memory image to cassette or disk
   0004
  0005
               * The user must load his own program in to cou

* cam. and then load this utility. He must put

* a DEF START.STOP in his program. This utility will

* prompt for a file name.
   8000
   8007 -
   8000
   0009
   0010
                 0011
                      REF VSBW.VSBR.VNBW.GFLLNK
REF DSRLNK.KSCAN
   DOM
   0013
               REF GRMWA.GRMRA.GRMRD.GPLWS
DEF SAVE
REF SFIRST.SLAST.SLOAD.VMER
   0014
   0015
   0016
   0017 2800
                      AORG >2800
   0018
                  TEXT (CC) COPYRIGHT 1981 BY TEXAS INSTRUMENTS, INC. (
   0019-2800 28
 0820 282E
 0057 2831
                      BYTE O
             CO
    0058 2832
                        EVER
    0059 2831 FILEN EQU $-1
0060 2832 FILE 898 32
                                      MUST BE ON EVEN WORDBOUNDARY
```

(11)

						.'								
SAVE		SOSMAC	3.4.0	81.11	7 08	8:54:13	MONDAY	/. JAN	25,	1982.				
				and the			);; »			ر بازاد افراد دارید افراد دارید		PAGE	0003	
.0061		2852	FILEND	EQU	\$			·	grand to		454			. :
0062						RST IS	LESS TH	AC MAR	iooo (			11.1	rejarrij	
						ST 1S I			7		* 5			
		20					* * * * * * * * * * * * * * * * * * * *							
0065						FILE N	AME			4 1 1 1	. **			
 0066	A	27 At 28 mg									J			
0067	2884	0600	CPAB	DATA	>0600	,>1000.	0 -		100			1000		o"
		2000												
		OF80												
0070		0F89								** **				
0071		OF8A	PABEN	EQU	>F80+	10	*							
0072	288E	0000	HEADER	DATA	0.0.0					*			*	
0073				EQU	>300									
0074	2BC4			EVEN										
0075	2804	87	INITS	BYTE	>87,>8	<,080,>	F.>00							
0076		2BC5	POS	EQU	INITS									
0077		2808	OFFS	EQU	<b>5-1</b>									
0078	2BCA			EVEN						· .				
0079	2BCA	87	0N	BYTE	>87.>	F								
0080	2800	00	CURSOR	BYTE	0,0,0	,0,0,0,	>70,>70	C C						
0081	2804	0000		DATA			ŕ							
												•		

```
SDSMAC 3.4.0 81.117 08:54:13 MONDAY, JAN 25, 1982.
 0083
0084
                     SCANS KEY BOARD FOR FILE NAME INPUT
 0085
 0086
0087
             * *----
               SAVE
 0088 2804
0088 2806 0400 CLR RO
 0090 2808 0201
                    LI R1.MENU
     280A 282E
 0091 2BDC 0202
                    LI R2.MENLEN
     280E 0300
 0092 2BEO 0420
               BLWP SVMBW
     2BE2 0000
 0093 28E4 0200 SAVE05 LI RO.>400
                                     SET UP SPRITE
      2BE6 0400
 0094 28E8 0201
                     LI R1.CURSOR
     28EA 28CC
 0095 28EC 0202
                    LI
                         R2.8
     28EE 0008
 0096 2BF0 0420
                     BLWP avmbw
     28F2 28E2
 0097 2BF4 C820
                     MOV WORLDINGTS TURN SPRITES ON
     2BF6 2BCA
     2BF8 2BC4
0099 2BFA
               LI R3, FADDR R3 -> CURRENT CHAR POSITION
 0100 28FA 0203
      2BFC 02E2
 0101 28FE C820
                 MOV BON, BINITS INITIALIZE SPRITES
 2CD0 2BCA 2CO2 2BC4
0102 2C04 SAV30
0103 2C04 06A0 BL ANSPRIT WRITE SPRITE
2C06 2C6C
0104 2C08 0420 BLWP &KSCAN WAIT FOR A KI
 2006 2060
0104 2008 0420
                     BLUP EKSCAN WAIT FOR A KEY
200A 0000
 0105 200C DOA0
                     MOVE aSTATUS, R2 - IS THIS A NEW KEY?
   200E 837C
 0106 2010 1603
                     JNE SAV40
JNP SAV30
B @DONE
                                      YES
 0107 2C12 10F8
                                      NO. WAIT
 0108 2014 0460 EXIT
     2C16 2DF8
 0109
                      CLR R1
                     MOVE aKEYCOD.R1 R1 -> KEY
 0112 2C1A 0060
      2010 8375
 0113 2C1E 0281
                     CI R1.>0800
                                      NO. BACK SPACE?
     2020 0800
 0114 2022 1317
                      JEQ BACKSP
                      CI R1.>0000 ENTER KEY?
 0115 2024 0281
      2026 0000
 0116 2028 132A
0117 202A 0281
                      JEQ ENTER
                     CI R1.>0500
      2020 0500
                      JEQ EXIT
  0118 2C2E 13F2
  0119 2030 0281
                      CI R1.>DF00
      2032 OF00
  0120 2034 13EF
                      JEG EXIT
  0121 2036 0281
                      CI 81.>2000
```

***							
				e at le ge			
,		2038	2000				
	0122	203A	1AE4		JL		
			0003	~/)	Von		WRITE KEY ON SCREEN
	0124	203E	0420		BLKP	avse4	
		2040	0000		4.4		
			0583		INC	R3 ·	BUMP CURSOR ADDRESS
-	0126		0283	9	CI	R3.24*32-1	
		2046	02FF		•		
	0127	2048	1808		JH	SAV20	
	0128	204A	8820		AB	aelght.apos	NOV SPRITE OVER
			2 C 5 C				
			2805				,
	0129	2050	1009		JMP	SAV30	NO. GET NEXT KEY
	0130	2052		BACKSP			-
	0131	2052	0283		CI	R3.FADDR	AT THE BEGINNING?
			02E2				
	0132	2056	1206		JLE	SAV30	YES, IGNORE KEY
	0133	2058	0403		DEC	R3	NO. BACK UP
	0134	205A	0201		Li	R1.>0890	
			0800				•
	0435		205 C	EIGHT	EQU	\$-2	
			7801		38	Ri.apos	ADJUST SPRITE -
		2060	2865				
	0137	2062	1000				GET REXT KEY
	0138	2064	0201	ENAME	LI	R1.ERR4	
		2066	2892	i i e i			
	0139	2068	0460		8	aE1010	
		2C6A	209A				
•	0140	2C6C		WSPRIT		4	
į.			0200		LI	RO,SAB	
		2C4E	0300				
	0142	2070	0201		LI	R1, INITS	أكرا المناسب والمالية
		2072	28C4			. Talaning Billion	المُنْفِينَا وَالْمُنْفِينِ وَالْمُنْفِينِ وَالْمُنْفِينِ وَالْمُنْفِينِينِ وَالْمُنْفِينِ وَالْمُنْفِينِ
			~~~	i.	LI	R2,5	
	and the second	2076	0202	, m	* *	10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	. 4144	ୁ∡୯୵ଅ	U42U :	177	BLWP	avmbw	
v	- Color Balling	207A	2BF2			**	
	0145	2C7C	D45B		RT		Control of the second
2	0146	207E		ENTER			
	0147	207E	0820		MOVB	aoffs, ainits	
		2080	2808	st in			•
, · .		2082	28C4				
'n	0148	2084	06A0	dag kati. Ma	81	awsprit	
			2060				
	0149		0200		LI	RO, FADOR	
			02E2			,	
	0150		0201		LI	R1.FILE	•
			2832				
	0151		0202		LI	RZ.32	
	<del>-</del>		0020		*	- r	
	0152		0420		BLWP	avmbr	READ FILE NAME
			0000	•		- · · · · <del>-</del> · · ·	73711.2700

SDSMAC 3.4.0 81.117 | DB:54:13 MONDAY, JAM 25. 1982.

PAGE 0005

SAVE

**(1)** 

SAVE	S05MAC 3.4.0	81.11	7 08:54:13	MONDAY. JAN 25, 1982.
		والمراكب		PAGE 0006
		" }		
0154				ng ang ago ago ago ang
0155				and the second s
0156	*	SCAN	FILE NAME FOR	R LENGTH
.0157	*			
0158				ENT START ADDRESS
0159	* *	M. W.	R7 -> CURRE	ENT #BYTES LEFT TO SAVE
0160	ang ang kabupa <b>*</b> 1 da ing	A Mich	[ R8j−> LAST	BYTE IN CPU FILE NAME BUFFER
0161	∕ ≯		R4 -> LENG	TH OF THE FILE NAME + 1
0162	*		1000	
0163	*	, ,		And also have have more made and made and state and state and state and state and the made and the state and the s
0164 2098	0203	LI	R3.F1LE	VA
	2832		4 - 1	
	0283 SAV45	CI	R3.FILEND	SCAN FOR END OF FILE NAME.
	2852			
0166 2CAC		JGT		
0167 2CA1		CE	*R3+.aBLANK	
2 CA4	286E			
0168 2CA6			SAV45	•
0169 2CA8	3 C203	MOV	R3,R8	R8 -> LAST SYTE IN FILE NAME
0170 2CAA	1 0548	DECT		
0171 2CA	0284	LI	R4,FILE	•
2CA5	2832			
0172 2080				CALCULATE LENGTH OF FILE NAME
0173 2082	2 0103	VOM	R3.R4	R4 -> # BYTES TO MOVE INTO VOP RAM
0174 2CB4	¥ 0603	DEC	R3	R3 = LENGTH BYTE
0175 2CB	5 0283	CI		
2CB	3 0003	rwine.		
0176 208/	4 11D4			BAD DEVICE NAME AND A CONTROL
0177 2CB 0178 2CB	C 06C3	SWPB	R3 .	
	E 0803	HOVB	R3, OFILEN	Harrier and the contract of th
zcci	J 2831			
0179	The second #			g and the company of
0180 200	2 0206	LI	R6,SLOAD	PROGRAM'S LOAD ADDRESS
2CC 0181 2CC	4 0000	V .		
0181 2CC	5 C806	nov	R6, aLOAD	
20 CC	3 2804	dr = -1	e det fa	
0182 200	A 0206	LI	R6.SFIRST	GET PROGRAM'S START ADDRESS
200	0000			
0183	*	CI	R6,>A000	VALID?
0184	· *	JĻ	ESTART	110
0185 2CC	E 0207	LI	R7,SLAST	GET PROGRAMS STOP ADDRESS
140 July 200	0 0000			
1 0186 2CD	2 61 C6 4 115B	S	R6,R7	FILE SIZE
0187 200	4 115B	JLT	ESTOP	BAD FILE SIZE

(

	SAUF		SUSMAC		84.44	7 88.54.17	MONDAY, JAN 25, 1982.
	JP V L		araiino		01.11		PAGE 0007
						* • •	
73	0189			*			y water tooks darke state stat
	0190			*		+ - 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	0191			*	MAIN	LOOP TO SAVE	EACH FILE IN MEMORY IMAGE FORMAT
	0192			*		CALCUATES FIL BUILDS HEADER	E SIZE
	0193		- 27	* .		BUILDS HEADER	
	0194			*		WRITES FAB	IF NEED TO WRITE ANOTHER FILE
			e transfer	*		CHECKS TO SEE	I IF NEED TO WRITE ANOTHER FILE
	0196			*			
-	- 0197.	~ A 6 /	0000	*			#BYTES TO MOVE
	0178	2006	CUS/	UWTU	nuv	87.82	TETICE TUNE OF S
	ULYY	2000	0287		UI	K1.* 5\(\frac{1}{2}\) \(\frac{1}{2}\) \(\frac{1}2\) \(\frac{1}{2}\) \(\frac{1}2\) \(\frac{1}2\) \(\frac{1}2\) \(\frac{1}2\) \(	BIGGER THAN 8K?
	0200	2008 2000	188 (440)	,	TIT	MAIN10 R2.>2000-6	MORE
	0200	2000	7102 0000		ا ساد دا	DO ZODODEN	1, (·) L
	± بند اسا		1FFA		ill.	Nathana M	
	0202			MATNIN	MOV	R2 . R4 O	R10 = # BYTES TO MOVE
						92.4	MAKE ROOM FOR HEADER
	است المناسدة السا		0006		F1 4	(, , , )	CEPSCOM (CONTORS) FOR SECURITY
	0204		C802		VON	R2.@FSIZE	
			288A		, , , , ,	, , , , , , , , , , , , , , , , , , ,	
	0205		0200		LI	RO.PAB	
			OFAC			***************************************	
	0206		0201		LI	R1.CPAB	
			2884				
- '	0207		0202		LI	R2.9	
		20F6	0009				
4	0208	2CF8	0420		BLWP	Warvs	WRITE 10 BYTES OF PAB
100			207A				
, X.							
			CO84		MOV	R4,R2	#BYTES IN FILE NAME
- 13 Mg 77			0200				
40.00			OF89				
100	0212	2002	0201	s fra	LI	R1 FILEN	MOVE, FILENAME INTO PAB
		2004	2831			Tanada in War	
5 - 5 - 5	0213	2006	0420		BLWP	avmbu	
1.00	444	2008	CZCFA D	agelel 12 Basila	01.0	RO	
- 1	MUZ14	20UA	_U+€U_{@	alanda in the second of the se	CLR	RU MENU	
H.	, uz15,	2000	0201	will.	LI.	R1.MENU	and the second of the second o
	0244	2040	282E -			DO MENHEN	RESTORE SCREEN
- 10 mg	the same of the				<u>i i</u>	IVAC 4 ETAIN TO A TO	NEW LONE OF REEN
F. T. Maria	0217	2014	. 0420 ±	F. I.	BIMP	avmbw	
		2014	2008	on Service	An 144 17 E	m # 111 m 31	
	0218	2D18	0200	•	LI	RO,SADDR	
		201A	0202			_ ,	
	0219	201C	0201		LI	R1.FILE	
	1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	201E	2832				
	0220	2020	0084		Von	R4,R2	
	0221	2022	0602		DEC	R2	
	0222	2024	0420		BLWP	avmbu	
		2026	2016				
	0223			*			
	0224	2028	04E0		CLR	<b>aheader</b>	•
			288E				
	0225	2020	81 CA		C	R10.R7	MORE DATA TO WRITE?
			1302			MAINZO	110
	0227		0720		SETO	<b>CHEADER</b>	YES, SET FLAG TO INDICATE THIS
			288E				
				MAIR20			
	0229	2034	0820		MOV	afsize.aHEAD	ER+2 LOAD FILE SIZE

\$75 \$25	SAVE		SDSMA	C 3.4.U	81.11	7 08:54:13	AL .YAGNOM 3	N 25, 1982.		#1 #4 #1 #4
									PAGE	กกกษ
<b>(5)</b>							garage and			
MC.	and the second second	2036 2038	28CO			and the second second			6,751.9	31.11
	0230	203A		ar al	Von	aLOAD, aHEADE	ER+4 L	OAD START ADD	RESS	
	W 62 64 44 .		2804							
		203E	2802							
	0231	2040	0200		LI	RO.>1000	era eksar		w., 1 & 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	and the second s		1000				in the Sheep of			The second second
	0232	2044			LI	R1, HEADER				
SON.	Apply that the state of	2048	288E		LI	20 A	•			77 475
d.m.	U233		0006		, L. J.	n u	•			
		2040			BLWP	SYMBN	WRITE HEAD	ER IN PABBUF		•
	3201		2026							
	0235			*						
	0235	2050	0200		ŁΙ	RO.>1006	WRITE AFTE	R THE HEADER		
		2052	1006					4		
	0237	2054	C08A		KOV	R10.R2	R10=SIZE	•		
£783.		2056			nov	R6.R1	R6=START	AN THEO USO		
	0239		0420		BLKP	avnew	MOVE PROGR	AN INTO VDP		
-			204E	<b>3</b> 4						
<b>(</b> )	0240		8820	*	С	afile.acs	CARRETTER			
****	0271		2832		v	di zmmtubb	and the same of the same of the same of			
. *			282E							
0	0242		1323		JEQ	CASSIO		** ** **		
	0243	2064		DISKIO						j.
4.1	0244	2064	0201		LI	R1,PABFL	DO A MEMOS	RY IMAGE SAVE	e de la companya de	Standard Contract
(4)			OF89							
			C801			R1.05CNAME		100 Å		an a
ASS.			8356			adsrlnk				
1237			0420		BLWF	aruskink :		7.0 (128-4)	regional property of	-1.16 8-41
	r yen Ye	2025	0000	Page 1998	DATA	g.	and the second second		North Tarte	
CB.			130F			E10	DISK ERRO	R TOTAL		
The second	1 To 1		0640	per extra contract to the cont	BL	1. The control of the			The west	
100	and the second second		2EOA		٠.	***	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
0		2078	.0060		NOV -	aHEADER, R1	VN agragi		Art Salar	
Marie III		207A	28BE					,		12/2
-41°-4			1330			DONE				
	V 1		A18A			R10.R6		NEW START	was significant	
Ž	and the second second second		61CA		S	R1U, K/	CALCULATE	NEW LENGTH	orae	The Control
			: A8QA : 2804		Α .	אזט,מבטאט	CHLUULHIE	NEW LOAD ADD		
. <b>₩⊅</b> *			8620		AB	aone,*R8	CHANGE F1	LE NAME		
in.			2830		170	W 20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				4
<b>©</b>			10A5		JMP	MAIN				
Št.	0257	•		*					17 .	+ #
	0258		0201		LI	R1,ERR2	BAD STOP	ADDRESS		
9			2851					1		•
			1004		JMP	E1010	D + 6 - 67 4 0 T	ANDERSE.		
6	0260			*START		R1.ERR1 EIO10	BAD START	nvuncaa .		
<b>10</b>	0261		ን ጠፋአጣ	* E10	JMP BL	ackerr				,
D	6252		2 06A0 4 2E0A		<i>□</i>	₩ W 1 \ E2 13 13				**
<b>6</b>	0263		5 0201		LI	R1,ERR3	10 ERROR	•		
**************************************	000		3 2872							
	0264		4 0200		LI	RO.EADDR				
(3)			0242			•				
	0249		E 0202		L I	R2.32				
A.		2DA9	3 0030							
at any										

								* * * * * * * * * * * * * * * * * * *		
SAVE		BOSMAC	3,4.0	81.11	7 08:54:13	MONDAY, J	AN 25,	1982.		
									PAGE	0009
0266	20A2 .	0420		BLWP	Wenve					
	20A4	205A								
0267		0460		8	@SAVEO5					
0268	ZUNG	28E4	*				,			4
0269			CASSIO		*					
0270		0201 % 0F80		LI	R1.PAB+13	POINT TO	END OF	"CS1"		
0271		0801		MOV	R1.8SCNAME			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		
	2080	8356				•				
0272		0201		LI	R1,3					
0273		C801		KOV	R1.2>8354	NAME IS 3	BYTES	LONG		
, m, m, m, s	2088									
UZ/4	20BC	0201 0800		LI	R1.>800	OFFSET FO	R SEAR	CH ·	•	
0275	208E			MOVE	R1.@>8360					
81-5-77	2000	836D 0200		LI	70 048CH	****** ******				
0276	2004			F. T	RO.PAEFN	MOAF AFAT	CE NAME	= IU FAC		
0277	2006	0201		LI.	R1.>834A				•	
0278	2008 200A			LI	99 T .					
. 0270		0003			112,3					
		0420		BLWP	avmer `					
	2000	2096	 *							•
0281	2002	04E0		CLR	2>83DO	CRULST, I	INITIAL.	IZE SEARC	4	
	2004	0.700			A 10			Control No. Warrante	,	
				HOLIE						ji kurili da da karana. Karana karana karan
0282	2006.	D820		ночв	a>8300,a>837					
0282	2006 2008 200A	D820 8300 8370			a>8300,a>837					
0282	2006 2008 200A 200C	D820 8300 8370 0420			@>8300,@>837					
0282 0283 0284	2006 2008 200A 200C 200E 200E	D820 8300 8370 0420 0000			agplink					
0282 0283 0284 0285	2006 2008 200A 200C 200E 200E	D820 8300 8370 0420 0000		BLWP DATA	agpLLNK >3D					
0282 0283 0284 0285 0286	2006 2008 200A 200C 200E 20E0	D820 8300 8370 0420 0000 0030		BLWP DATA	agplink					
0282 0283 0284 0285 0286	2006 2008 200A 200C 200E 20E0 20E2 20E2 20E4 20E6	D820 8300 837C 0420 0000 0030 04A0 2E0A C060		BLWP DATA BL	agpLLNK >3D					
0282 0283 0284 0285 0286 0286	2006 2008 200A 200C 200E 20E0 20E2 20E2 20E4 20E6 20E8	D820 8300 8370 0420 0000 0030 06A0 2E0A C060 2BBE		BLWP DATA BL HOV	aGPLLNK >3D ackerr aheader.R1	FINISHED'				
0282 0283 0284 0285 0286 0286	2006 2008 200A 200C 200E 20E0 20E2 20E4 20E6 20E8 20EA	D820 8300 837C 0420 0000 0030 04A0 2E0A C060		BLWP DATA BL MOV JEQ	aGPLLNK >3D ackerr aheader.r1 done	FINISHED YES -				
0282 0283 0284 0285 0286 0287 0287 0289	2006 2008 200A 200C 200E 20EO 20E2 20E4 20E4 20E8 20EA 20EC 20EE	D820 8300 8370 0420 0000 0030 06AD 2E0A C060 2B8E 1306 A18A A80A		BLWP DATA BL MOV JEQ A	aGPLLNK >3D ackerr aheader.R1	FINISHED' YES - UPDATE ST				
0282 0283 0284 0285 0286 0287 0288 0289 0290	2006 2008 200A 200C 200E 20EO 20E2 20E4 20E6 20EA 20EA 20EC 20EE 20FO	D820 8300 8370 0420 0000 0030 06A0 2E0A C060 2BEE 1306 A18A A80A 2804		BLWP DATA BL MOV JEQ A	aGPLLNK >3D  ackerr aheader.r1  Done R10.R6 R10.aload	FINISHED YES - UPDATE ST	CART = 1			
0282 0283 0284 0285 0286 0287 0289 0290 0291 0292	2006 2008 200A 200C 200E 20EO 20E2 20E4 20E6 20EA 20EC 20EC 20EC 20FC 20FC 20FC	D820 8300 8370 0420 0000 0030 0640 2060 2886 1306 A18A A80A A80A 610A 6400		BLWP DATA BL MOV JEQ A A	aGPLLNK >3D ackerr aheader.r1 DONE R10.R6 R10.aload	FINISHED  YES  UPDATE ST	CART = 1	LEFT		
0282 0283 0284 0285 0286 0287 0289 0290 0291 0291	2006 2008 200A 200C 200E 20EO 20E2 20E4 20E6 20EA 20EA 20EC 20FO 20F2 20F4 20F6	D820 8300 8370 0420 0000 0030 0640 2060 2886 1306 A18A A2004 6104 0460 2006		BLWP DATA BL MOV JEQ A A S B	aGPLLNK  >3D  aCKERR  aHEADER.R1  DONE R10.R6 R10.aLOAD  R10.R7 amain	FINISHED YES UPDATE ST	FART START	LEFT		
0282 0283 0284 0285 0286 0287 0289 0290 0291 0292	2006 2008 2008 2000 2000 2000 2000 2000	D820 8300 837C 0420 0000 0030 06AD 26AD 26AD 28BE 1306 A18A A80A 2804 61CA 0460 2CD6	#	BLWP DATA BL HOV JEQ A A S B	aGPLLNK  >3D  ackerr  aheader.r1  bone  R10.R6  R10.R7  amain	FINISHED' YES - UPDATE ST	FART STATES	LEFT		
0282 0283 0284 0285 0286 0287 0289 0290 0291 0292	2006 2008 2008 2000 2000 2000 2000 2000	D820 8300 837C 0420 0000 0030 06AD 26AD 26AD 28BE 1306 A18A A80A 2804 61CA 0460 2CD6	#	BLWP DATA BL HOV JEQ A A S B	aGPLLNK  >3D  ackerr  aheader.r1  bone  R10.R6  R10.R7  amain	FINISHED' YES - UPDATE ST	FART STATES	LEFT		
0282 0283 0284 0285 0286 0287 0289 0290 0291 0292	2006 2008 200A 200C 200E 20EO 20E2 20E4 20E6 20E8 20EC 20EC 20F2 20F4 20F6 20F6 20F8 20F8 20F8	D820 8300 8370 0420 0000 0030 0640 2060 2886 1306 A18A A80A 2804 610A 0460 2006	#	BLWP DATA BL HOV JEQ A A S B	aGPLLNK  >3D  ackerr  aheader.r1  bone  R10.R6  R10.R7  amain	FINISHED' YES - UPDATE ST	FART STATES	LEFT		
0282 0283 0284 0285 0286 0287 0289 0290 0291 0292 0293 0294 0295 0296	2008 2008 2008 2000 2000 2000 2000 2000	D820 8300 837C 0420 0000 06AD 26AD 26AD 26AD 26AD 26AD 26AD 26AD 2	* DOME	BLWP DATA BL MOV JEQ A A CLR MOVB	aGPLLNK >3D ackerr aheader.R1 DONE R10.R6 R10.aload R10.R7 amain RU R0.a>837C	FINISHED' YES - UPDATE ST	FART STATES	LEFT		
0282 0283 0284 0285 0286 0287 0287 0290 0291 0292 0293 0294 0295 0296	2008 2008 2008 2000 2000 2000 2000 2000	D820 8300 8370 0420 0000 0630 0640 2604 2886 4184 4804 6104 6104 6104 6104 6104 6104 6104 61	* DONE	BLWP DATA BL MOV JEQ A A S B CLR MOVB LWF1	aGPLLNK  >3D  ackerr  aheader.r1  DONE R10.R6 R10.aload  R10.R7 amain  RU R0.a>837C  GPLNS	FINISHED' YES - UPDATE ST	EYTES	LEFT	eten over tody solve expe	have sales
0282 0283 0284 0285 0286 0287 0287 0290 0291 0292 0293 0294 0295 0296	2008 2008 2008 2000 2000 2000 2000 2000	D820 8300 837C 0420 0030 06A0 2E0A0 2E0A0 2E0A0 480A A80A A80A 2B0A 61CA 04C0 04C0 04C0 04C0 04C0 04C0 04C0 0537C 04C0 0537C	* DONE	BLWP DATA BL MOV JEQ A A S B CLR MOVB LWF1	aGPLLNK  >3D  ackerr  aheader.r1  DONE R10.R6 R10.aload  R10.R7 amain  RU R0.a>837C  GPLNS	FINISHED' YES - UPDATE ST	EYTES	LEFT	eten over tody solve expe	have sales
0282 0283 0284 0285 0286 0287 0289 0290 0291 0292 0293 0294 0295 0296 0297	2008 2008 2008 2000 2000 2000 2000 2000	D820 8300 837C 0420 00030 06A0 260A0 260A0 2886 418A 2804 61CA 0460 2CD6 04CD 0837C 02CD 03CD 03CD 0002	# DONE	BLWP DATA BL MOV JEO AA SB CLR MOVB LWF1 LIMI	aGPLLNK  >3D  ackerr  aheader.r1  DONE R10.R6 R10.aload  R10.R7 amain  RU R0.a>837C  GPLWS 2	FINISHED' YES - UPDATE ST	EYTES (	LEFT	eten over tody solve expe	have sales
0282 0283 0284 0285 0286 0287 0289 0290 0291 0292 0293 0294 0295 0296 0297	2008 2008 2008 2000 2000 2000 2000 2000	D820 8300 837C 0420 0030 06A0 2E0A0 2E0A0 2E0A0 480A A80A A80A 2B0A 61CA 04C0 04C0 04C0 04C0 04C0 04C0 04C0 0537C 04C0 0537C	# DONE	BLWP DATA BL MOV JEO AA SB CLR MOVB LWF1 LIMI	aGPLLNK  >3D  ackerr  aheader.r1  DONE R10.R6 R10.aload  R10.R7 amain  RU R0.a>837C  GPLNS	FINISHED' YES - UPDATE ST	EYTES (	LEFT	eten over tody solve expe	have sales
0282 0283 0284 0285 0286 0287 0287 0289 0290 0291 0292 0293 0294 0295 0297 0298	2008 2008 2008 2000 2000 2000 2000 2000	D820 8300 8370 0420 0030 0640 02060 2008E 1306 4804 4804 6104 6100 0400 0400 0400 0400 0400 04	* DONE	BLWP DATA BL MOV JEO AA SB CLR MOVB LWF1 LIMI	aGPLLNK  >3D  ackerr  aheader.r1  DONE R10.R6 R10.aload  R10.R7 amain  RU R0.a>837C  GPLWS 2	FINISHED' YES - UPDATE ST	EYTES (	LEFT	eten over tody solve expe	have sales

08:54:13 MONDAY, JAN 25, 1982. S0SMAC 3.4.0 81.117 0302 2E0A 0200 3 LI RO.PAB+1 STATUS BYTE 2E0C 0F81 0303 2E0E 0420 BLRP avser READ 1 BYTE FROM VDP 2E10 0000 RIGHT JUSTIFY ERROR CODE nov R1.R1 0305 2E14 CO41 ERROR=0 JEQ: CKRT 4.5 0306 2E16 1306 0307 2E18 06C1 SWPB R1 R1,>3000 CONVERT ERROR TO ASCIL 0308 2E1A 0221 AI 2E10 3000 MOVE R1.3ERR3+11 PUT IN ERROR MESSAGE 0309 2E1E 0801 -2620 2670 JMF E1005 0310 2E22 10B9 8.7 0311 2E24 0455 CKRT END 0312 NO ERRORS.

	SAVE						7 08	154:43	MONDAY	. JAK	<u>25</u> , 19	82.		0044
	LABEL	- V	ALUE	D1	EFR	REFERE	INCES						PAGE	0011
	¢.		2E26			0042	0045	0052	0059	0061	0077	0135	- :.	
	BACKSE		2052		0130	0114								
	BLANK		256E	100		0167								
	CASSIO		2DAA		0269	0242								
	CKERR		2EOA		0301	0249	0262	0286			*			
	CHRT		2E24	1	0311	0306	5.00							1. 1. A.
	CPAB		2884		0067	0206			in which t	7. ·		2 × 1		
	05		2625		0055	0241								
	CURSOR		28 C C		0380	0074					+ 4			
	018810		2064		0243	<i></i>	C	in ma						
	DONE		20F8		0294	0108	0251	0288						
	OSRLUK		204E 0242		0013 0045	0245								
	EADDR		2050		0135	0128								
	EIO		2092			0248								
	E1005		2095		0263	0310								
	E1010		209A		0264	0139	0259							
	EMAME		2064			0165	0176					•		•
	ENTER		207E		0145	0116								
	ERR2		2852		8688	0258								
	ERR3		2872		0064	0263	0309		•		,			
	ERRA		2292		0055	0138								
	ERRCOD		2882		0065	m - A m								
	ESTOP	:	2080		0256 0108	0187 0118								. •
	EXIT -		2014 02E2			0100		0149						
	FILE		2832			0150		0171	0219	0241				
3.7	FILEM	en magena Se de la compensa	2831			0178								
	FILEND	275	2852		0061	0165	1 1 h					- 2		
	FSIZE.	6	2BBA						Solve .		447	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	in in the second	
	GPLLNK					0283			estrance in a			180	a programme	
	GPLHS		\$E00	1 1997 1993		0297		1 S May 1 1				1. 4.22		BV & THE
	GRMRA:		مسأبي الما		.0014. 0014.	talijita		1000	11 1			tyropy yezhoù e K	لىلىنىڭ ھالار ئەرىكىنىڭ	All the second second second
Ş,	GRMWA	: R : R			0014			1		1 m	مهام مواج جا برد	<b>美国的</b>	where we we	
	HEADER		2886		0072	n224	0227	0229	0230	0232	0250	0287		
Ċ	INITS		2804				0097						The second second	San
	KEYCOD		8375			0112							And the second second	The grant of the second of the
	KSCAN	R	200A		0013	0104							4	
	LOAD		2804		0081		0230	0254	0290					
	MAIN		2006		0198		0292							A Property of the Control of the Con
	MAIN10	1 1 1				0200	and the second second	and the second						
	MAIN20 MENLEN		·2034 -0300		0052	0226 0091			•				1.152.153	
	MENU		282E		0026			0052	0095	0215	*			46 11 A
	OFFS		2808		0077	0147		to the sales.						
	ON		2BCA		0079	0097								
	ORE		2630		0056	0255								
	PAB		OF80		0069	0205	0270	0302						
	PABFL		0F89		0070	0211	0244							
	PABFA		OFSA		0071	0275								
4	PADDR		0202		0053									
	POS		28 CS		0075	0128								
	80		0000			0089			0141	0149	0205	0211	0214	0218
	54		(**) 3* 6 1 5 4			0231		0254	0276	0295	0296	0362	Cra a Ci	0404
	R1		0001			0090 0134		0111 0133	0112 0141	0113 0150	0115 0206	0117 0212	0119 0215	0121 0219
						0232		0244	0142	0250	0258	0263	0215	0217
						9272		3174	0275	0277	0257	0304	0305	0305
						0307		0309						

**(E)** 

6)

(in

**(1)** 

(L.)

**(**(3)

0

0

O

3

(3)

SAVE	SUS VALUE	MAC 3.4.6	, 81.11 REFERE		:54:13	MONDA	Y, JAN	25. 1	982.	1	E 0012
LAREL .	VALUE	Dark	NEFERE	ичца				1. 1			L 0012
R10 .	000A		0202	0225	0237	0252	0253	0254	0289	0290	0291
R2	0002		0091	0095	0105	0143	0151	0198		0202	0203
*1 **		¥*	0204	0207	0210	0216	0220	0221	0233	0237	0265
			0278								*
83	0003	17	0100	0123	0125	0126	0131	0133	0164	0165	0157
			0167	0172	0173	0174	0175	0177	0178	1	
R4	.0004		0171	0172	0173	0210	0220	1.6		100	
R &	10004		0188	0181	0182	01£6	0236	0252	0269		
R7	0007		0185	0185	0198	0199	0225	0253	0291		,
RB	9968		0169	0170	0255						
SAB	0300	.0073	0141								
SADDR	0202	0042	0218								
SAV20	2854	0099	0127								
SAV30	2004	0102	0107	0122	8129	0132	0137				
SAV40	2018	0110	0106								
SAV4S	2090	0165	0168								
SAVE	D 2806	8800	0015								***
SAVECS	28E4		0267								
SCHAME	- 8354			0271							*
SFIRST											
SLAST	R 2000										
SLOAD	R 2004										
STATUS	8370										
VMBR	R 2000			0279			6\15. A 77	0.04.75	6000	0.534	(m, m, m, c)
VMBW	R 20A4	0012	0092	8200	U144	0208	0213	0217	0222	0234	0239
	مدرسوي يس	. :	0268								
VSBR	R 2E10			eroje e i i .							
VSEW	R 2040		0124	· 04/3	- 1						12 12 14 17 12 13 17 18
	2060	0140			tara da Sala			41.45			e e e e e e e e e e e e e e e e e e e

Ð