

ALPHA MICRO USERS SOCIETY

NEWSLETTER

VOL. I NUMBER 5

JUNE 1978

ABOUT THIS NEWSLETTER

The AMUS Newsletter seems to be coming out every five weeks as opposed to monthly. This is largely due to late breaking stories and a lot of good information arriving late concerning available software.

Starting with this issue, we will begin publishing page-long descriptions of AMUS members' software, hardware, consultation services, training services, or what-have-you pertaining to Alpha Micros. To ease the load on the Newsletter staff, please print all information on one page, as clearly and as concisely as possible. The material sent in by RMC Computer Services in this Newsletter is a good example of the kind of thing we would like to see.

JOHN FRENCH LEAVES ALPHA MICRO

We have received word that John is no longer the Vice President of Marketing. He is still a member of the Board of Directors.

For questions about sales, marketing, or deliveries, you can contact Patrice Martinez, Bob Hitchcock, or Dennis Tofte.

Dick Wilcox requests that everyone have a little patience during this transition period.

WORD PROCESSOR

Come on folks, who has one? Everybody wants one and is nagging us about it. Jim Taylor has written up the specs for a word processor, and if no one comes up with one by the next issue of the Newsletter, we will publish the specs and see who salutes it.

### AMUS SEMINAR

We would like to put together a seminar for new A/M owners and persons considering an Alpha Micro. Each AMOS program would be covered, along with terminal driver modification, implementing AMS format, new SYSTEM.INI's, and Memory Management. Workshops would also be held on use of the MACRO, the EDITOR, TXTFMT, and ALPHABASIC with examples and hands on practice. Tentative plans call for the seminar (at least the first one) to be held in Boulder in September. If you are interested in participating, please let us know.

### WANCO FLOPPIES

If you have a Wanco, please let us know, so that we can put you in touch with other users.

### CORRECTION

Last newsletter stated that sequential files need contiguous area on the disk. Obviously, everyone knows that we should have said random files.

### TAPE DRIVE

If you have interfaced a tape drive to the A/M, please let us know.

### SOFTWARE

Several folks have promised to send information on software that we have not received. Please send anything you have to AMUS so that we can get it into the newsletter. AMUS is also making up a disk of software that will be available for the cost of the disk to AMUS members.

### 10Mbyte DISKS

The 10 Megabyte disks are being delivered...got yours yet?

### LOAD, DEL

In AMOS version 3.3 you cannot load LOAD.PRG. However, you can delete DEL.PRG.

### IF THEN

Be careful using multiple statement lines. Especially with the IF THEN command in BASIC. The following statements give erroneous results:

```
10 A=2
20 IF A=1 THEN 30 : PRINT A
30 PRINT "WRONG"
```

RUN

WRONG

### IF THEN (CONT.)

Also, the command line IF A=1 THEN 200 ELSE 300 will give SYNTAX ERROR, however IF A=1 THEN 200 ELSE GOTO 300 will work.

### BASORT - 3.3

The polyphase merge portion of the BASORT does not work. The SORT does work if all records will fit into memory.

### EBCIDIC

We have a requirement for the ASCII to EBCIDIC conversion for a floppy disk to be read by IBM equipment. If you have any ideas, please send them to AMUS

### PRINT COMMAND

The AMOS version of the print spooler does not function with disk names other than DSK as mentioned in the last newsletter. However, courtesy of COMPUTER WORKSHOP of Northern Virginia, we have a listing of both the LINE PRINTER SPOOLER and PRINT SPOOLER REQUEST programs with the modifications that will allow the use of the PRINT command with any disk format. Those listings are included in this newsletter.

### REQUESTS

There should be a way to prevent or recover from the problems caused by the mounting and unmounting of disks by users who do not inform other users of the disk change.

There is a need to define a real security system for the A/M, or at least define documentation about what AMOS will and won't do within it's system security.

There is a need for a command that will print leading zeros within the PRINT USING command. The use of a Ø in place of # would be handy.

### PERSCI

PERSCI has a kit available that attaches head #1 more securely.

When the mainframe power is turned off and the power is still on at the PERSCI disk drive, it is possible for the PERSCI to receive extraneous commands from the mainframe. On occasion, this command turns out to be a seek track 97, which causes the head to bang into the end of the shaft and can cause head #1 to loosen. To prevent this, always turn off the disk drive before turning off the mainframe.

### 3.4 RELEASE

A/M says that AMOS 3.4 should be out by the end of June. It will include:

- An ISAM that works
- PASCAL
- BASIC fixes

## CONVERTING FROM STD TO AMS FORMAT

### Step One:

Obtain the PPN's on all the disks you wish to convert to AMS format

### Step Two:

1. Rename PERAMS.DVR to AMS.DVR
2. Change SYSTEM.INI as follows:
  - A. Insert SYSTEM DSKØ:AMS.DVR [1,6] before the SYSTEM command
  - B. Change DEVTEL; remove DSK1 and replace it with AMS1
  - C. Change BITMAP DSK,32,1 to BITMAP AMS,39,1
  - D. Change MOUNT DSK1: to MOUNT AMS1:
3. Reboot the system

### Step Three:

1. Place a blank disk into the DSK1 position in teh disk drive. This position is now referred to as AMS1: so you now have DSKØ: and AMS1:
2. Run AMSFMT (you must be logged in under DSKØ: [1,4]). The drive number is 1
3. After formatting the disk, log into DSKØ: 1,2 and SYSACT AMS1: Initialize the disk and add the appropriate PPN's
4. Log onto AMS1: under one of the new PPN's
5. COPY its files from DSKØ: to AMS1: using:  
COPY /X=DSKØ: [p,PN] \*  
Do this for each PPN you wish to have on AMS1:

To create a system disk in the AMS format do all of the above, copying teh PPN's one at a time. Your original SYSTEM.INI must be changed to show the BITMAP size of 39 as in step two. MONGEN must be used to create a new SYSTEM.MON with PERAMS.DVR as the driver. To bring the system up inder control of your new format, if you do not have a Version D 2708 chip, bring the system up using a standard formatted disk. Load PERLOD.PRG. Switch disks, place the AMS formatted system disk in DSKØ:. Type PERLOD and it will load up your new system.

### FREE SCHOOL PUBLISHES STARTER AMOS MANUAL

The Community Free School of Boulder, Colorado has finally finished its new, improved version of a user's manual for AMOS. The manual is meant as an aid for new Alpha Micro owners who don't know all the things that AMOS assumes you know. Included in the manual are step-by-step directions for formatting new disks, and for sharing main memory and disk space. Use of the EDITor is discussed with examples, and suggestions are included to avoid fatal errors.

The manual should be an excellent introduction for dealers to have on hand for prospective customers, since it discusses what the system can and cannot do, and gives various timings for computations and file handling activities for dedicated and heavily shared systems.

The manual is available from the Free School for \$7.50 plus \$.75 mailing. Orders may be sent

Attn: Steve Elliott  
Community Free School  
P.O. Box 1724  
Boulder, Colorado 80306

### HISTORY OF AMOS

The following pages were sent to AMUS by Guruprem Singh Khalsa of Khalsa Computer Systems, Inc. of Pasadena, California. They trace the development of AMOS from version 0.0 through 3.3 and show the number of blocks that each AMOS program occupied for each version. We have also included a blank page of the form that Guruprem used so that you can keep track of the upcoming releases of AMOS.

Next newsletter, we'll include the history of the drivers, command files, LISP programs, and sources.

### ALPHA MICRO SCORES ON SAN LUIS OBISBO REPORT

AMUS recently came across a report done by an independent agent that rated a dozen mini and micro systems for ease of programming, multi-user features, cost, service, and other items. Alpha Micro came out tops over an IBM system costing \$38K, a DEC system at about \$30K and four other systems more expensive than the \$14K A/M system. Its a twenty page report so there isn't room to reproduce it here. Alpha Micro plans to publish it and make it available to dealers soon, but if you can't wait, let us know and perhaps we can arrange to get it copied for you.

PROGRAM	VERS 0.0	VERS 0.1	VERS 0.2	VERS 0.3	VERS 1.0	VERS 1.01	VERS 1.02	VERS 2.0	VERS 3.0	VERS 3.1	VERS 3.2	VERS 3.3	COMMENT
DSKØ: 1,4													
AMS .MON								1					see AMS.DVR
AMSCPY.PRG								1	1	1			
AMSFMT.PRG								1	1	1	1	1	
AMSLOD.PRG								1	1				
AMSORT.PRG												4	
APPEND.PRG									1	1	1	1	
ASCDMP.PRG								1	1	1	1	1	
ATTACH.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
BASIC .PRG	17	15	15	16	19	19	19	20	21	21	22	22	
BASIC1.INI			1										
BASIC1.MON			33										
BASIC2.INI			1										
**													
BASIC2.MON			47										
BASORT.SBR												2	
BAUD .PRG												1	
BDT .PRG	1			1	1	1	1						
BITMAP.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
BMVR .PRG	1	1	1	1	1	1	1	1	1	1	1	1	
CITY .PRG		1	1										
CLEAR .PRG	1	1	1	1	1	1	1						
CLKFRQ.PRG				1	1	1	1	1	1	1	1	1	
COMPILE.PRG												18	
COPY .PRG	2	2	2	2	2	2	2	2	2	2	2	2	
COUNT .PRG	1	1	1	1	1	1	1	1	1	1	1	1	
**													
DDT .PRG	11	11	11	11	11	11	11	11	11	11	11	11	
DEL .PRG	1	1	1	1	1	1	1	1	1	1	1	1	
DETACH.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
DEVTBL.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
DIAG1 .PRG				1	1	1	1	1	1	1	1	1	
DIAG2 .PRG												1	
DIAG3 .PRG												2	
DING .PRG				1	1	1	1	1	1	1	1	1	
DIR .PRG	3	3	3	3	3	3	3	3	3	3	3	3	
DIRSEQ.PRG									2	2	2	2	



PROGRAM	VERS 0.0	VERS 0.1	VERS 0.2	VERS 0.3	VERS 1.0	VERS 1.01	VERS 1.02	VERS 2.0	VERS 3.0	VERS 3.1	VERS 3.2	VERS 3.3	COMMENT
DSKØ: 1,4		**			**			**			**		
LOG .PRG	2	2	2	2	2	2	2	2	2	2	2	2	
LOGOFF.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
LPTSPL.PRG	2	2	2	2	2	2	2	2	2	2	2	2	
MACRO .PRG	16	16	16	16	16	16	16	16	16	16	16	16	
MAC1 .OVR	1	1	1	1	1	1	1	1	1	1	1	1	
MAC2 .OVR	4	4	4	4	4	4	4	4	4	4	4	4	
MAC3 .OVR	9	9	9	9	9	9	9	10	10	10	10	10	
MAKE .PRG	1	1	1	1	1	1	1	1	1	1	1	1	
MAP .PRG	2	2	2	2	2	2	2	2	2	2	2	2	
MEMDEF.PRG									1	1	1	1	
MEMORY.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
MEMPLT.PRG									1				
		**			**			**			**		
MEMIST.PRG		1	1										see DIAG1
MONGEN.PRG									1	1	1		
MOUNT .PRG									1	1	1	1	
PACK .PRG	1	1	1	1	1	1	1						
PDLFMT.PRG									7	7	7	7	
PERSCI.MON				20	20	20	20	22					see MONGEN
PERLOD.PRG								1	1	1	1	1	
PRINT .PRG	1	1	1	1	1	1	1	1	1	1	1	1	
QDT .PRG	1	1	1	1	1	1	1	1	1	1	1	1	
QUEUE .PRG										1	1	1	
RAND .PRG	1	1	1	1	1	1	1						
REDALL.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
		**			**			**			**		
RENAME.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
REVIVE.PRG	1	1	1	1	1	1	1	1	1	1	1	1	
RNDRED.PRG	1	1	1	1	1	1	1		1	1	1	1	
RUN .PRG		12	12	12	16	16	16	17	18	19	19	21	
SAVE .PRG	2	2	2	2	2	2	2	2	2	2	2	2	
SCREEN.SBR									1	1	1	1	see ADM3.TDV SOROC.TDV
SEEK .PRG	2	2	2	2	2	2	2						
SEND .PRG	1	1	1	1	1	1	1						
SET .PRG								1	1	1	1	1	
SETNUL.PRG					1	1	1						







LINE PRINTER SPOOLER

CHANGE 1: THIS CHANGE WILL ALLOW THE USE OF THE "PRINT" COMMAND WITH FILES FROM ANY DISK FORMAT AND NOT JUST "DRL." THIS CHANGE IS COURTESY OF THE COMPUTER WORKSHOP OF NORTHERN VIRGINIA.

COPY SYS

QSIZE=10 ;NUMBER OF ENTRIES IN QUEUE TABLE

EXTERN

;THE FIRST 2 WORDS OF LPTSPL ARE PREDEFINED FOR COMMUNICATION

LSTS: ;WORD 0 ;STATUS FLAGS  
;WORD QUEUE ;OFFSET TO QUEUE TABLE

JMP INIT ;INITIAL ENTRY JMP

;SLEEP UNTIL MARKED BY PRINT CALL

LPTSPL: MOV @JCNCR,R0 ;SET JOB INDEX  
JWAIT J.EW ;ENTER EXTERNAL WAIT STATE  
LCNCR: DEASGN LDB,R5 ;INSURE PRINTER NOT ASSIGNED  
LEA @M,QUEUE ;INDEX THE QUEUE  
TST @M ;QUEUE EMPTY?  
BEO LPTSPL ; YES

;WAIT FOR 500 NS TO ALLOW FOR POSSIBLE PRINTER DEPEND FROM ANOTHER JOB  
;ASSIGN THE PRINTER UNLESS SOMEONE ELSE HAS IT IN WHICH CASE WE WAIT

LCNRT: SLEEP \$30. ;SLEEP FOR 500 NS  
ASSIGN LDB,R5 ;ATTEMPT TO ASSIGN THE PRINTER  
BNE LCNRT ; BUT SLEEP WHILE IF IN USE

;LPT HAS BEEN ASSIGNED - SET REQUEST FROM QUEUE INTO DDB

MOV @M,R5 ;SAVE QUEUE BASE  
MOV (R4)+,DDB+20 ;MOVE DDC-CD TO DDB \*\*\*CHANGE 1\*\*\*  
MOV (R4)+,DDB+24 ;MOVE FILENAME TO DDB  
MOV (R4)+,DDB+26  
MOV (R4)+,DDB+30  
MOV (R4)+,DDB+32 ;MOVE FPN  
MOV (R4)+,DDB+22 ;MOVE DRIVE

;OPEN THE FILES

LCORR: CLRS DDB+34 ;INSURE NOT OPEN  
OPEN DDB ;OPEN DISK INPUT FILE  
BNE LCNEL ; NOT FOUND  
MOV @R2,R3 ;SAVE LINK  
OPENO LDB ;OPEN LPT OUTPUT  
BIS @1,LSTS ;SET ACTIVE STATUS

;BOTH FILES ARE OPEN - TRANSFER THE DATA

LCPR: BIT @2,LSTS ;CHECK STATUS FOR KILL  
BNE LCON  
MOV R3,@R2 ;RESTORE LINK  
INPUT DDB ;READ RECORD  
BNE LCON ; ERROR  
TST DDB+4 ;EOF?  
BEO LCON ; YES  
MOV @R2,R3 ;SAVE LINK  
CLR @R2 ; AND CLEAR TO NULL

MOV DDB+4,DDB+4 ;SET BYTE COUNT  
CONV LDB ;SET THE OFF DATA  
BR LCON

;END OF FILE - CLOSE UP SHOP

LCOR: CLOSE DDB ;CLOSE DDB  
CLOSE LDB ;CLOSE LPT  
BIC @1,LSTS ;CLOSE ACTIVE AND KILL FLAGS

;DELETE THE REQUEST BY SHIFTING ENTIRE QUEUE TABLE DOWN

LCDEL: MOV @R0+,@R5+ ;SHIFT DOWN  
BEO LCON ; BELLOW END  
MOV @R0+,@R5+  
MOV @R0+,@R5+  
MOV @R0+,@R5+  
MOV @R0+,@R5+  
MOV @R0+,@R5+  
BR LCON

DDB: WORD 41400 ;DDB END

BLEN 7  
BLEN 13

LDB: WORD 41400 ;LPT END

LCOR: BLEN 23  
QUEUE: BLEN 12 ;QUEUE \*\*\*CHANGE 1\*\*\*  
WORD 0,-1 ;END OF QUEUE NUMBER

;THE FOLLOWING AREA COMPOSES THE INITIALIZATION CODE

;IT IS ALSO THE DATA BUFFER AREA WHICH IS USED ONCE THE SPOOLER IS RUNNING

DEF: WORD 0 ;BUFFER SIZE  
BWT: CLR LSTS ;CLEAR STATUS WORD  
MOV @R0L,@R5+2 ;SET QUEUE TABLE OFFSET  
SPEC @R2,TW ;SPEC THE PRINTER NUMBER  
LEA @R2,BUF ;INDEX THE BUFFER  
MOV @R2,DDB+2  
MOV @R2,DDB+2  
JOBSET @J.LPT,JOBSET ;SET LPT SPOOLER JOB TYPE  
MOV @R2+20,@R5+24 ;TRANSFER SERVICE MODE TO PETCH  
PETCH LPTSPL  
BEO LDB  
PETCH @R5+24 ;LPT. END NOT FOUND  
TYPCR < NOT FOUND  
EXIT

LDB: USRFR R1 ;CALCULATE SIZE OF REQUIRED PARTITION

USRFR R0  
SUB @R1  
JOBSET R1,JOBSET ;SHOWING THE PARTITION  
JMP LPTSPL ;START SPOOLER  
LPTWR: BLEN 12 ;PETCH END

RADR0 /LPT DDB/  
WORD 006  
;BUF+510. ;REST OF BUFFER  
WORD 0  
END

```

*****
;
; PRINT SPOOLER REQUEST PROGRAM
;
*****
;CHANGE 1: THIS CHANGE WILL ALLOW THE USE OF THE "PRINT"
;          COMMAND WITH FILES FROM ANY DISK FORMAT AND NOT
;          JUST "DSK.". THIS CHANGE IS COURTESY OF THE
;          COMPUTER WORKSHOP OF NORTHERN VIRGINIA.
;
*****

```

```
COPY SYS
```

```
EXTERN
```

```

;SCAN JOB TABLE TO LOCATE SPOOLER JOB
RUNN:  MOV    @#JOBTAB,R0    ;SET JOB TABLE INDEX
SCAN:  BIT    #J.LPT,JOBTYP(R0) ;SPOOLER JOB?
      BNE    GOTCHA        ; YUP
      ADD    @#JOBESZ,R0    ;KEEP LOOKIN'
      TST   @R0            ; UNLESS END OF TABLE
      BNE    SCAN
      TYPECR (LPTSPL NOT ALLOCATED)
      EXIT

;SPOOLER IS ACTIVE - ENTER REQUEST IN QUEUE TABLE
GOTCHA: MOV   JOBBAS(R0),R4   ;INDEX THE PROGRAM
      ADD   #ZID,R4         ; WHICH ASSUMES LPTSPL IS FIRST
      BYP
      LIN
      BEQ   QLST            ;DEFAULT OF /L
      CMPB @R2,#'/'        ;CHECK FOR LIST SWITCH
      BNE   GNK
      CMPB 1(R2),#'L
      BEQ   QLST
GNK:   LOCK                ;LOCK PROCESSOR
      FSPEC DHB,LST         ;GET FILE SPEC TO PRINT
      LEA  R1,DHB+32       ;INDEX THE PPN
      TST @R1              ;WE MUST SET OUR OWN IF DEFAULT
      BNE .+10
      JOBGET @R1,JOBUSR
      LEA  R1,DHB+22       ;INDEX THE DRIVE
      CMP @R1,#377        ;WE MUST SET OUR OWN IF DEFAULT
      BNE .+22             ; ***CHANGE 1***
      JOBGET @R1,JOBDRV
      LEA  R1,DHB+20       ; ***CHANGE 1***
      JOBGET @R1,JOBDEV    ; ***CHANGE 1***
      BYP                ;CHECK FOR KILL SWITCH
      CMPB @R2,#'/'
      BNE   GNK2
      CMPB 1(R2),#'K
      BEQ   QKTL
GNK2:  INIT   DHB
      LOOKBP DHB
      BEQ   LOOKOK
      PFILE  DHB
      TYPECR < NOT FOUND>
      EXIT
LOOKOK: ADD 2(R4),R4       ;BUMP TO THE QUEUE TABLE
QLP:   CMP 2(R4),#-1      ;END OF TABLE?
      BEQ QEND           ; YUP
QEND

```

```

*****
;
; LINE PRINTER SPOOLER
;
*****
;
; CHANGE 1: THIS CHANGE WILL ALLOW THE USE OF THE "PRINT"
; COMMAND WITH FILES FROM ANY DISK FORMAT AND NOT
; JUST "DISK." THIS CHANGE IS COURTESY OF THE
; COMPUTER WORKSHOP OF NORTHERN VIRGINIA.
;
*****

```

```

COPY SYS

QSIZE=10 ;NUMBER OF ENTRIES IN QUEUE TABLE

EXTERN

;THE FIRST 2 WORDS OF LPTSPL ARE PREDEFINED FOR COMMUNICATION
LSTS: WORD 0 ;STATUS FLAG
;WORD QUEUE ;OFFSET TO QUEUE TABLE

JMP INIT ;INITIAL ENTRY JMP

;SLEEP UNTIL AWAKENED BY PRINT CALL
LPTSPL: MOV @AJPCUR, R0 ;SET JOB INDEX
;WAIT J.E.W ;ENTER EXTERNAL WAIT STATE
LCMCK: DEASN LDB, R5 ;ENSURE PRINTER NOT ASSIGNED
LEA R4, QUEUE ;INDEX THE QUEUE
TST @R4 ;QUEUE EMPTY?
BEO LPTSPL ; YES

;WAIT FOR 500 NS TO ALLOW FOR POSSIBLE PRINTER DENIED FROM ANOTHER JOB
;ASSIGN THE PRINTER UNLESS SOMEONE ELSE HAS IT IN WHICH CASE WE WAIT
LCMRT: SLEEP $30. ;SLEEP FOR 500 NS
ASSIGN LDB, R5 ;ATTEMPT TO ASSIGN THE PRINTER
BNE LCMRT ; BUT SLEEP WHILE IF IN USE

;LPT HAS BEEN ASSIGNED - SET REQUEST FROM QUEUE INTO LDB
MOV R4, R5 ;SAVE QUEUE BASE
MOV (R4)+, DDB+20 ;MOVE DVC-CD TO DDB ***CHANGE 1***
MOV (R4)+, DDB+24 ;MOVE FILENAME TO DDB
MOV (R4)+, DDB+28
MOV (R4)+, DDB+30
MOV (R4)+, DDB+32 ;MOVE FPN
MOV (R4)+, DDB+22 ;MOVE DRIVE

;OPEN THE FILES
LCOEN: CLMB DDB+34 ;ENSURE NOT OPEN
OPEN DDB ;OPEN DISK INPUT FILE
BNE LCOEL ; NOT FOUND
MOV @R2, R3 ;SAVE LINK
OPENO LDB ;OPEN LPT OUTPUT
BIS $1, LSTS ;SET ACTIVE STATUS

;BOTH FILES ARE OPEN - TRANSFER THE DATA
LCPRF: BIT $2, LSTS ;CHECK STATUS FOR KILL
BNE LCOEN
MOV R3, @R2 ;RESTORE LINK
INPUT DDB ;READ RECORD
BNE LCOEN ; ERROR
TST LDB+4 ;EOF?
BEO LCOEN ; YES
MOV @R2, R3 ;SAVE LINK
CLR @R2 ; AND CLEAR TO NULLS

```

```

MOV DDB+4, DDB+4 ;SET DVC CD
OUTPR LDB ;SET PRINTER OFF
BR LCMRT

;END OF FILE - CLOSE UP SHOP
LCMCK: CLOSE DDB ;CLOSE DDB
CLOSE LDB ;CLOSE LPT
BIC $1, LSTS ;CLEAR ACTIVE AND KILL FLAG

;DELETE THE REQUEST BY SHIFTING ENTIRE QUEUE TABLE DOWN
LDBL: MOV @R0+, @R0+ ;SHIFT DDB
MOV @R0+, @R0+ ; UNLESS DDB
MOV @R0+, @R0+
MOV @R0+, @R0+
MOV @R0+, @R0+
MOV @R0+, @R0+
BR LDBL

DDB: WORD 41400 ;DVC DDB
;RND 7
;RND /RND/
;RND 13
LDB: WORD 41400 ;LPT DDB
;RND 23
QUEUE: RND 12. *QSIZE ;REQUEST QUEUE ***CHANGE 1***
;RND 0, -1 ;END OF TABLE NUMBER

;THE FOLLOWING AREA COMPRISES THE INITIALIZATION CODE
;IT IS ALSO THE DATA BUFFER AREA WHICH IS USED ONCE THE SPOOLER IS RUNNING
INIT: WORD 0 ;SETUP DVC
CLR LSTS ;CLEAR STATUS WORD
MOV @RNDL, LSTS+2 ;SET QUEUE TABLE OFFSET
RSPCL LDB, RND ;SPEC THE PRINTER NUMBER
LEA R2, RND ;INDEX THE BUFFER
MOV R2, DDB+2
MOV R2, DDB+2
JOBSET $1, LPT, JOBSET ;SET LPT SPOOLER JOB TYPE
MOV LDB+20, LPTDDB+24 ;TRANSFER BUFFER NAME TO FPN
FPCON LPTSPL
BEO LFG
FPCON LPTSPL+24 ;LPT.DDB NOT FOUND
TYPCR < NOT FOUND>
EXIT
LFG: USRVE R1 ;CORRECT SIZE OF REQUESTED PARTITION
USRVS R0
SUB R0, R1
JOBSET R1, JOBSET
JMP LPTSPL ;START SPOOLER
LPTDDB: RND 12 ;FETCH DDB
;RND /LPT DDB/
;RND 406
;DDB+10, ;REST OF BUFFER
;RND 0
END

```

\*\*\*\*\*  
 PRINT SPOOLER REQUEST PROGRAM  
 \*\*\*\*\*

CHANGE 1: THIS CHANGE WILL ALLOW THE USE OF THE "PRINT"  
 COMMAND WITH FILES FROM ANY DISK FORMAT AND NOT  
 JUST "DSK." THIS CHANGE IS COURTESY OF THE  
 COMPUTER WORKSHOP OF NORTHERN VIRGINIA.  
 \*\*\*\*\*

COPY SYS

EXTERN

```

;SCAN JOB TABLE TO LOCATE SPOOLER JOB
RUNN:  MOV    @#JOBTAB,R0    ;SET JOB TABLE INDEX
SCAN:  BIT    #J.LPT,JOBTYP(R0) ;SPOOLER JOB?
      BNE    GOTCHA        ; YUP
      ADD    @#JOBESZ,R0    ;KEEP LOOKIN'
      TST   @R0            ; UNLESS END OF TABLE
      BNE    SCAN
      TYPECR (LPTSPL NOT ALLOCATED)
      EXIT

;SPOOLER IS ACTIVE - ENTER REQUEST IN QUEUE TABLE
GOTCHA: MOV    JOBBAS(R0),R4    ;INDEX THE PROGRAM
      ADD    #ZID,R4        ; WHICH ASSUMES LPTSPL IS FIRST
      BYP
      LIN
      BEQ    QLST          ;DEFAULT OF /L
      CMPEB @R2,#'/        ;CHECK FOR LIST SWITCH
      BNE    GNK
      CMPEB 1(R2),#'L
      BEQ    QLST
GNK:   LOCK
      FSPEC  DHB,LST        ;LOCK PROCESSOR
      LEA   R1,DHB+32      ;GET FILE SPEC TO PRINT
      TST  @R1            ;INDEX THE PPN
      BNE  .+10           ;WE MUST SET OUR OWN IF DEFAULT
      JOBGET @R1,JOBUSR
      LEA   R1,DHB+22      ;INDEX THE DRIVE
      CMP  @R1,#377       ;WE MUST SET OUR OWN IF DEFAULT
      BNE  .+22           ; ***CHANGE 1***
      JOBGET @R1,JOBDRV
      LEA   R1,DHB+20      ; ***CHANGE 1***
      JOBGET @R1,JOBDEV    ; ***CHANGE 1***
      BYP
      CMPEB @R2,#'/
      BNE    GNK2
      CMPEB 1(R2),#'K
      BEQ    QKIL
GNK2:  INIT  DHB
      LOOKUP DHB
      BEQ   LOOKOK
      PFILE DHB
      TYPECR < NOT FOUND>
      EXIT
LOOKOK: ADD    2(R4),R4    ;BUMP TO THE QUEUE TABLE
      QLP:  CMP    2(R4),#-1 ;END OF TABLE?
      BEQ   QEND        ; YUP
  
```

```

TST @R4 ;EMPTY ENTRY?
BEQ QOK ; YES
ADD #14,R4 ;NEXT QUEUE ENTRY ***CHANGE 1***
BR QLP

QEND: TYPECR [NO ROOM IN QUEUE]
QLST3: EXIT
;FOUND EMTY QUEUE ENTRY - PUT IN REQUEST
QOK: MOV DHB+20,(R4)+ ;PUT IN DVC-CD ***CHANGE 1***
MOV DHB+24,(R4)+ ;PUT IN NAME ***CHANGE 1***
MOV DHB+26,(R4)+
MOV DHB+30,(R4)+
MOV DHB+32,(R4)+ ;PUT IN_PPN
MOV DHB+22,@R4 ;PUT IN DRIVE
BIC #177400,(R4)+
JRUN J.EXM ;WAKE UP SPOOLER
EXIT

;LIST SWITCH PROCESSING
QLST: LOCK ;LOCK PROCESSOR
MOV #[DSK],DHB+20
BIT #1,@R4
BNE QLST1
TYPECR [SPOOLER IDLE]
QLST1: ADD 2(R4),R4 ;BUMP TO QUEUE TABLE
QLST2: TST @R4 ;END OF TABLE?
BEQ QLST3 ;YES
MOV (R4)+,DHB+20 ;MOVE DVC-CD TO DHB ***CHANGE 1***
MOV (R4)+,DHB+24 ;MOVE ENTRY TO DHB
MOV (R4)+,DHB+26
MOV (R4)+,DHB+30
MOV (R4)+,DHB+32
MOV (R4)+,DHB+22
PFILE DHB
CRLF
BR QLST2

;KILL SWITCH PROCESSING
QKIL: MOV R4,R3 ;SAVE SPOOL STATUS FLAG ADDRESS
MOV @R4,R5 ;R5 BIT 0 IS SET IF CURRENT OUTPUT MUST BE KILLED
ADD 2(R4),R4 ;BUMP TO QUEUE TABLE
QKIL1: TST @R4 ;END OF TABLE?
BEQ QKIL5 ;YES, NOT FOUND
CMP DHB+20,@R4 ;COMPARE ENTRY WITH SPEC***CHANGE 1***
BNE QKIL4 ; ***CHANGE 1***
CMP DHB+24,2(R4) ; ***CHANGE 1***
BNE QKIL4 ;
CMP DHB+26,4(R4) ; ***CHANGE 1***
BNE QKIL4 ;
CMP DHB+30,6(R4) ; ***CHANGE 1***
BNE QKIL4 ;
CMP DHB+32,10(R4) ; ***CHANGE 1***
BNE QKIL4 ;
BIT #1,R5 ;CANCEL CURRENT OUTPUT?
BEQ QKIL2 ;NO, JUST REMOVE FROM QUEUE
BIS #2,@R3 ;YES, SET CANCEL BIT
EXIT

QKIL2: MOVI 5,R3 ;SLIDE DOWN REMAINTNG ENTRIES
QKIL3: MOV 14(R4),(R4)+ ;MOVE DOWN ONE ENTRY ***CHANGE 1***
SOB R3,QKIL3
TST @R4 ;END OF TABLE?
BNE QKIL2 ;NO, CONTINUE
EXIT

QKIL4: ADDI 14,R4 ;MISMATCH, ADV NXT ENTRY***CHANGE 1***
CLR R5 ;AND CLEAR ACTIVE FLAG

```

```
BR QKIL1
```

```

QKIL5: PFILE DHB
TYPECR < IS NOT IN THE QUEUE>
EXIT
DHB: BLAW 24
END

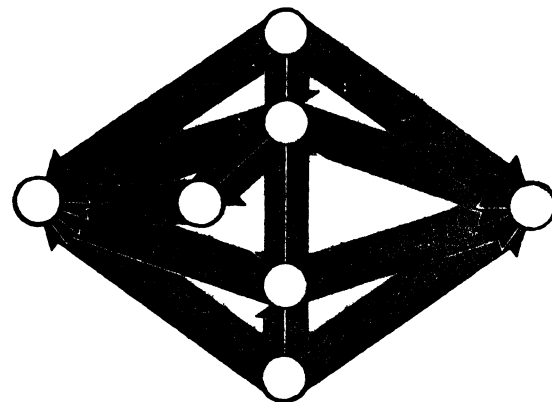
```

These are the A/M Assembler Mnemonics. Numbers refer to A/M manual page numbers

ADC	22	ADD CARRY	MBBA	26	MOVE BLOCK OF BYTES TO ADDRESS
ADD	30	ADD	MBBD	26	MOVE BLOCK OF BYTES DOWN
ADDI	14	ADD IMMEDIATE	MBRU	26	MOVE BLOCK OF BYTES UP
AND	31	AND	MBWA	26	MOVE BLOCK OF WORDS TO ADDRESS
ASH	28	ARITHMETIC SHIFT	MBWD	25	MOVE BLOCK OF WORDS DOWN
ASHC	28	ARITHMETIC SHIFT COMBINED	MBWU	25	MOVE BLOCK OF WORDS UP
ASL	19	ARITHMETIC SHIFT LEFT	MOV	32	MOVE
ASLB	23	ARITHMETIC SHIFT LEFT BYTE	MOV8	33	MOVE BYTE
ASR	19	ARITHMETIC SHIFT RIGHT	MOVI	15	MOVE IMMEDIATE
ASRB	23	ARITHMETIC SHIFT RIGHT BYTE	MSKO	6	MASK OUT
BCC	12	BRANCH IF CARRY CLEAR	MUL	29	MULTIPLY
BCS	13	BRANCH IF CARRY SET	NEG	20	NEGATE
BEQ	11	BRANCH IF EQUAL TO ZERO	NEGB	24	NEGATE BYTE
BGE	11	BRANCH IF GREATER THAN OR EQUAL TO ZERO	NOP	1	NO OPERATION
BGT	12	BRANCH IF GREATER THAN ZERO	PRTN	6	POP STACK AND RETURN
BHI	12	BRANCH IF HIGHER	RESET	1	I/O RESET
BHIS	12	BRANCH IF HIGHER OR SAME	REST	3	RESTORE REGISTERS
BIC	31	BIT CLEAR	ROL	18	ROTATE LEFT
BICI	15	BIT CLEAR IMMEDIATE	ROLB	22	ROTATE LEFT BYTE
BIS	31	BIT SET	ROR	18	ROTATE RIGHT
BISB	33	BIT SET BYTE	RORB	22	ROTATE RIGHT BYTE
BIT	32	BIT TEST	RRTN	4	RESTORE AND RETURN FROM SUBROUTINE
BLE	12	BRANCH IF LESS THAN OR EQUAL TO ZERO	RRTT	4	RESTORE AND RETURN FROM TRAP
BLO	13	BRANCH IF LOWER	RSTS	4	RESTORE STATUS
BLOS	12	BRANCH IF LOWER OR SAME	RSVC	4	RETURN FROM SUPERVISOR CALL
BLT	12	BRANCH IF LESS THAN ZERO	RTN	6	RETURN FROM SUBROUTINE
BMI	12	BRANCH IF MINUS	RTT	4	RETURN FROM TRAP
BNE	11	BRANCH IF NOT EQUAL TO ZERO	SAVE	3	SAVE REGISTERS
BPL	12	BRANCH IF PLUS	SAVS	3	SAVE STATUS
BPT	3	BREAKPOINT TRAP	SBC	22	SUBTRACT CARRY
BR	11	BRANCH UNCONDITIONALLY	SDLA	17	SHIFT DOUBLE LEFT ARITHMETIC
BVC	12	BRANCH IF OVERFLOW CLEAR	SDLR	16	SHIFT DOUBLE LEFT ROTATE
BVS	12	BRANCH IF OVERFLOW SET	SDRA	17	SHIFT DOUBLE RIGHT ARITHMETIC
CLR	19	CLEAR TO ZEROS	SDRR	16	SHIFT DOUBLE RIGHT ROTATE
CLRB	23	CLEAR BYTE TO ZEROS	SET	19	SET TO ONES
CMP	32	COMPARE	SETB	23	SET BYTE TO ONES
CMPB	32	COMPARE BYTE	SDB	28	SUBTRACT ONE AND BRANCH (IF NOT=0)
COM	20	COMPLEMENT	SSLA	16	SHIFT SINGLE LEFT ARITHMETIC
COMB	24	COMPLEMENT BYTE	SSLR	15	SHIFT SINGLE LEFT ROTATE
DEC	20	DECREMENT	SSRA	15	SHIFT SINGLE RIGHT ARITHMETIC
DECB	24	DECREMENT BYTE	SSRR	15	SHIFT SINGLE RIGHT ROTATE
DIV	29	DIVIDE	SSTS	21	STORE PROCESSOR STATUS
FADD	36	FLOATING POINT ADD	SUB	31	SUBTRACT
FCMP	37	FLOATING POINT COMPARE	SUBI	14	SUBTRACT IMMEDIATE
FDIV	37	FLOATING POINT DIVIDE	SVCA	9	SUPERVISOR CALL A
FHUL	37	FLOATING POINT MULTIPLY	SVCB	9	SUPERVISOR CALL B
FSUB	36	FLOATING POINT SUBTRACT	SVCC	9	SUPERVISOR CALL C
HALT	2	HALT	SWAB	20	SWAP BYTES
IAK	6	INTERRUPT ACKNOWLEDGE	SWAD	23	SWAP DIGITS
IDS	2	INTERRUPT-DISABLE	SXT	21	SIGN EXTEND
IEN	2	INTERRUPT ENABLE	XCALL	21	TABLED SUBROUTINE CALL
INC	20	INCREMENT	TJMP	21	TABLED JUMP
INCB	24	INCREMENT BYTE	TST	19	TEST WORD
IW2	21	INCREMENT WORD BY TWO	TSTB	22	TEST BYTE
JSR	27	JUMP TO SUBROUTINE	WFI	3	WAIT FOR INTERRUPT
LCC	8	LOAD CONDITION CODES	XCH	28	EXCHANGE
LEA	27	LOAD EFFECTIVE ADDRESS	XCT	2	EXECUTE SINGLE INSTRUCTION
LSTS	21	LOAD PROCESSOR STATUS	XOR	31	EXCLUSIVE OR
MABB	26	MOVE ADDRESS TO BLOCK OF BYTES			
MABW	26	MOVE ADDRESS TO BLOCK OF WORDS			



# RMC Computer Services



June 10, 1978

Alphamicro Software

## GENERAL LEDGER SYSTEM

This general ledger system will support account #'s up to 8 digits, multiple companies, user control of asset total etc, any company year-end, program asisted month-end and year-end, automatically retains transactions for next month until the current month is closed, capability to print a simulated posting report of all or only selected transactions for the month.

## FINANCIAL REPORTS

This subsystem allows the operator to define the report format, then the reporting procedure. This approach allows a user to define one or more of his reports in more than one format or style. This subsystem works with the general ledger described above.

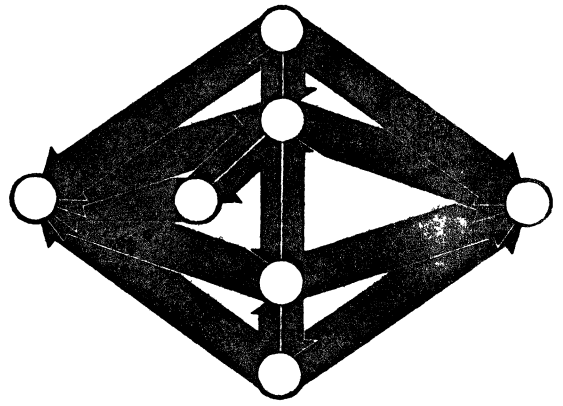
## ACCOUNTS PAYABLE SYSTEM

This system provides the user with manual & void checks, invoice discounts, periodic contract payments, invoice payment through operator authorization only, and a detailed report of all invoices both paid and unpaid by vendor. (paid invoice data limited by file size) Invoice entry and check writing generate automatic entries to the G/L system.

## PLOT

This program written in BASIC accepts X-Y point data from a sequential file, scales the data to fit user specified paper size, then outputs a plot to either a printer or another sequential file. The user has enormous control over his plot from axis annotation and titles to control of what portion of the input data gets plotted. Available through a limited license for \$50.(source)

# RMC Computer Services



Alphamicro Software continued

## MAILING LIST

This mailing list system uses cursor control on either the ADM3 or SOROC terminal. Each file entry has space for name, company, street, city, state, zip, & phone #. You may sort on any field and print either a roster or labels. Source available through a limited license for \$50.

## DEMO PACKAGE

This package consists of eight games and two demo programs. All operate from a central demo-menu program. Program names are: FLIGHT, HANGMA, HILO, LOAN, LUNAR(with cursor control), MASTER(mind), QUBIC, SUB, TREK(enhanced), & WHIZ. Source available for \$25.

## NATURAL LANGUAGE DATA BASE MANAGER

Natural language systems have been recognized for some years as a powerful and practical method for data-base management with large-scale computers in government, military, and scientific applications. The Lawndale Data-Base Management System (LAWMAN) brings the power of natural language processing to the microcomputer field, where it can be put to use in many more down to earth applications.

Implemented for the Alpha Microsystem AM-100 Computer, processing of files is controlled by sentences written in English, rather than by programs written in some computer language. For example, in our store, we often ask the computer to "List mailing whose interest is business systems by Zip Code as labels".

Definitional capability allows easy categorization of records into terms meaningful to the user. For example, telling the computer "DEF: Pasadena Residents : Mailing whose Zip Code is between 91101 and 91108 but not 91102" would allow subsequent processing using only the words "Pasadena Residents" to refer to that group.

Help messages are available throughout the system for user convenience. By simply entering a question mark, at any time, the system will respond with a message specifying the type of input LAWMAN is expecting. Dual question marks will print the message in greater detail.

A new approach to memory management optimizes the mapping of data, and the loading of programs, for the amount of memory available in the user's system. This is done automatically, and is transparent to the user. Thus thousands of records can be stored, ordered, categorized and otherwise processed in a matter of seconds, rather than in minutes or hours. Practical applications include:

Mailing lists	Real Estate listings
Market analysis	Quality Control Records
Inventory files	Telephone directory
Personnel files	Reservation system
Customer listing	Mail order management
Buyers Guide	Library cross indexing
Budget Analysis	Appointment scheduling
Price lists	Employment agency file

A ready to use disk, with complete user's manual, is priced at \$1250. Manual only, \$5.00. Enhanced versions, as they become available, will be supplied to all user's for a copying charge of \$25.00, including revised manual.

For more information, call or visit the Byte Shop of Lawndale. We're 3 miles South of the Los Angeles International Airport, and three blocks South of the San Diego Freeway.

BYTE SHOP OF LAWNSDALE  
16508 Hawthorne Blvd.  
Lawndale, CA 90260  
(213) 371-2421

# **AM-100 GENERAL LEDGER**

The Kitzmiller Systems General Ledger is part of a comprehensive business accounting system for the Alpha Microsystems AM-100 computer with floppy disk drives. Other parts of the total system are Accounts Receivable, Accounts Payable, Inventory Control, and Sales Analysis.

The system comes from seventeen programs of over two thousand total lines of Alphabasic computer programming code. This division into the program modules is done so that small businesses can operate with only 48,000 characters of memory.

Minimum hardware configuration includes the AM-100 computer, 48K of memory, dual floppy disk drives, a CRT and a printer. Larger businesses may require more than 48K of memory.

## **EASE OF OPERATION**

The system gives simple instructions which guide the user through the entire accounting process. The user does not have to be an expert in either accounting or computers.

To run the system the user simply enters the command RUN GL. Then the computer presents a set of "menu" options to the user.

## **REPORTS**

The Kitzmiller Systems General Ledger System prints the following reports:

- Chart of Accounts Listing
- General Journal Transactions
- Cash Payments Journal Transactions
- Cash Receipts Journal Transactions
- Trial Balance
- Profit & Loss (Income) Statement
- Capital (Net Worth) Statement
- Balance Sheet

The user has the option to print these reports on either his terminal or the system printer.

# PROGRAMS

**GL** - This is the main General Ledger program. It guides the user into the particular action he wishes to perform.

**GLCHAR** - The General Ledger Chart of Accounts program. A Chart of Accounts is a list of the many ways that a business can categorize its assets, liabilities, net worth, income and expenses. This allows the user to add, delete, and change information in the Chart of Accounts. The user has the option to print the Chart of Accounts on either his terminal or the system printer.

**GLCP JR** - The General Ledger Cash Payments Journal. This allows you to enter data of all transactions where your business spends money. The person entering data does not have to be familiar with the Chart of Accounts for your business. She can enter the fact that a check was written to Joe Smith. Later, in a procedure called posting, a person more familiar with the finances of the business can tell the computer which account in the Chart of Accounts such as professional fees, corresponds to the expense for Joe Smith. This procedure greatly speeds and simplifies data entry. The user has the options to add transactions to the Cash Payments Journal, post the cash payment transactions, and print the Cash Payments Journal on either the user terminal or the system printer.

**GLCRJR** - The General Ledger Cash Receipts Journal. This allows you to enter data of all transactions where your business takes in money. As before, the person entering the data does not have to be familiar with the Chart of Accounts for your business. She enters the fact that a certain amount of income came from a sale to Smedley Supply Company. Later, in the posting process, a person with more familiarity of the business finances can tell the computer which account corresponds to the sale to Smedley Supply Company. The user has the options to add transactions to the Cash Receipts Journal, post the transactions, and print the Cash Receipts Journal on either the user terminal or the system printer.

---

**GLJRNL** - The General Ledger General Journal Entry program. The General Journal is used when it is not appropriate to enter transactions through either the Cash Payments Journal or the Cash Receipts Journal. As with the other journals the person entering the data does not have to be familiar with the name or account numbers in the Chart of Accounts. The user has the option to enter transactions or print the General Journal on either the user terminal or the system printer.

**GLMECL** - General Ledger Month End Closing program. Month end closing is an accounting procedure where temporary accounts in the Chart of Accounts (income accounts and expense accounts) are reset to zero and the net amount is transferred to the net worth accounts of the owners. GLMECL handles the part of resetting the income and expense accounts to zero.

**GLMEDI** - General Ledger Month End Distribution program. This program transfers the net earnings to the capital (net worth) accounts of the owners of the business. This can be done for up to twenty partners in the business with the user selecting the amount of increase to be allotted to each partner.

**GLMEDR** - General Ledger Month End Closing Drawing Distribution program. This program adjusts the net worth of each of the partners by the amount that he has taken from (drawn) from the business or put into the business. This applies to any additional amount that is not considered to be salary. As before this can handle up to twenty partners in the business.

**GLMEMA** - General Ledger Month End Closing Main Program. This program guides the user through each phase of month end closing.

**GLMEST** - General Ledger Month End Closing Setup program. This program sets up the computer so that it knows which accounts are involved in the month end closing procedure.

**GLPOST** - General Ledger Posting program. This program allows the user who is familiar with the Chart of Accounts for the business to inform the computer of the account numbers corresponding to each of the various transactions entered by a less knowledgeable person.

**GLREPT** - General Ledger Report Printing Program. This program allows the user to print various reports on either his user terminal or the system printer. Reports are Monthly Income (Profit & Loss) Statement, Year to Date Income Statement, Capital Statement (shows increase in net worth for the month), and Balance Sheet (shows where all the assets, liabilities, and net worth are distributed). Programming optimization resulted in the savings of about 400 lines of computer programming in this program.

**GLSETU** - General Ledger Report Setup program. This program allows the user to make his reports to any way he wants to. The user can set up his own Income Statement, Capital Statement, and Balance Sheet. This action needs to be done only once.

**GLTRIB** - General Ledger Trial Balance program. This program prints out a Trial Balance, a list of all debits (assets and expenses) and all credits (liabilities, net worth, and income). The debits must equal the credits.

The system also includes several other programs for maintenance of the entire General Ledger System.

For more information contact your dealer or:

**KITZMILLER SYSTEMS**  
252 South Oxford Avenue  
Los Angeles, CA 90004  
Phone: 385-9388

ALPHA MICRO USERS GROUP  
P.O. BOX 1723  
Boulder Colorado 80306

# Quill computer systems ltd.

1605 NELSON STREET, VANCOUVER, BC. V6G 1M3

10 th May 1978

Dear Jim,

QUILL COMPUTER SYSTEMS is proud to announce the immediate availability of the following Systems Software :

- A. PASCAL Language Compiler....This version of BRINCH-HANSON's implementation was transferred from the original PDP 11/45 public domain source to create a seven pass, incremental, AMOS compatible, multi-user package that is more efficient than the UCSD PASCAL that also will run on AM-100 systems in the future release from ALPHA MICROSYSTEMS.DOCUMENTED fully
- B. Z-80 CROSS ASSEMBLER uses macro statements defining the standard ZILOG mnemonics to allow a text file to be created with the EDIT command and generate Z-80 object from source code. Self documented source supplied. BYTESAVER is required to burn EPROMS using integral BURN program for multiple 2708's
- C.8080 CROSS ASSEMBLER as above allows micro development using INTEL mnemonics . Self documented source supplied.
- D. Fairchild F-\*8 (3850) CROSS ASSEMBLER uses macros which define mnemonics used by Fairchild and second sources allows object code to be generated from source created with the EDIT command. Self documented source supplied.
- E. RCA 'COSMAC' 1802 CROSS ASSEMBLER as above uses RCA mnemonics for CMOS micro development via the EDIT command.  
Self documented source supplied.
- F. MICRO-TERM character generation program for ACT IV terminals. GRAPHIC design of symbols, APL, line drawing made easy.

### MORE SOFTWARE

Micheal's Realty	313 557-0770	Universal Data Entry and Update
Basic Business Software	801 531-1795	Math Packages
Ken Martin	801 486-5447	Games
Gene Erekson	801 375-6310	Startrek
Dick Leach	303 279-7796	Driver for DIT Terminal
Stan Viet	212 686-7923	PDQ Electric Memo (Data Base System) Renumbr Basic Retail Inventory Program
Bruce Zimmer	213 371-2421	Data Base Mgmt System
Guruprem Khalsa	213 684-3313	Communications Package
Jack Hatfield	503 773-8162	Data Mgmt Package

We will publish all of the software information that we have along with any description we have in the July issue.

### EXCERPTS FROM A LETTER

In answer to your questions on the PERSCI Errors, there are two schools of thought: One is that the 6 pots in the back of the PERSCI are all synchronized at the factory and you shouldn't mess with them. The other is that turning at least the front pot has solved some problems and I haven't heard that it has created any.

Head #1 is fastened to head 0 only by a set screw and can become loosened and get out of line with head 0 especially if seek track 97 command is issued (there are only 77 track on the disk and the head bangs into the end). This can happen occassionally if the power to your mainframe is shut off before the power to the PERSCI. You can loosen the set screw, move the head, tighten the set screw, try it, repeat if necessary.

It appears that the front pot can be turned about 10 full turns and the drive will still work so it should be centered at about 5 turns each way. If you SET DSKERR and run REDALL on each drive you may find that the setting is not the same for both drives.

If you have an oscilloscope, you should be able to get the exact information you need from PERSCI. Call Fred Richardson at (213) 820-3764.

### NEXT NEWSLETTER

Deadline is July 15. Those of you that missed the June software cutoff can get them in the July newsletter.

## BASIC BUSINESS SOFTWARE OFFERS FILE HANDLING PROGRAMS

Peter Burke of Basic Business Software in Salt Lake City sent us the following information on their file handling programs. For more information, contact him directly at: PO Box 2032, Salt Lake City, Utah 84110.

BBS is offering the programs as a package for \$15.00 by itself, and for \$22.50 with the Book "Some Common Basic Programs". Price includes shipping.

1. CREATE - This program will allocate a random file given the number of records desired, and the size of each record. It allows a practice run without actually allocating space which calculates wasted space for optimum use of disk space. It may be run in compiled mode.

2. RFTOSF - This program converts a random file to a sequential file, and will KILL the random file if desired. Since MAP statements are used, the program must be loaded in BASIC and then RUN. There are prompts within the program for changing the MAP statements.

3. SFTORF - This program converts a sequential file to a random file. The random file need not be allocated at run time, since this is automatically done by the program. as with RFTOSF, there are prompts for changing MAP statements, and the program must be loaded in BASIC to be run.

4. RFTORF - This program converts the size of a random file, automatically allocating the space for the new file. This is especially helpful for expanding the logical record size of a random file.

5. SORTER - This program sorts either random or sequential files. After prompting for names, keys, etc., it chains to a command file which loads the necessary subroutines for sorting. It may be run in compiled mode.

NOTE: All sequential files are saved with the extension of .SDF, and all random files are saved with the extension .RDF. INPUT files are assumed to have the same extension.



SOFTWARE PACKAGE NO. 1

=====

THIS PACKAGE CONSISTS OF 76 PROGRAMS WHICH APPEARED IN OSBORNE & ASSOCIATES BOOK "SOME COMMON BASIC PROGRAMS", WHICH WAS WRITTEN BY LON POOLE AND MARY BORCHERS. THE PROGRAMS ARE BROKEN UP INTO FOUR DIFFERENT AREAS:

- (1) STATISTICS
- (2) FINANCE
- (3) MATH
- (4) MISCELLANEOUS

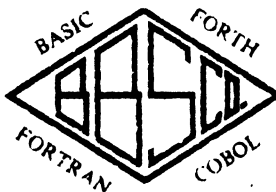
THIS PACKAGE IS USEFULL TO MANY INDIVIDUAL NEEDS FOR IT COVERS A VERY BROAD RANGE OF APPLICATIONS. ON THE FOLLOWING PAGES IS A LIST OF ALL THE PROGRAMS WHICH ARE INCLUDED IN THIS PACKAGE.

THE PROGRAMS WERE ORIGINALLY WRITTEN IN WANG BASIC, BUT WE HAVE CONVERTED THESE BASIC PROGRAMS TO THE INDIVIDUAL BASIC LANGUAGES THAT WE HAVE LISTED BELOW FOR SALE. ALL PACKAGES INCLUDE ONE OR MORE FLOPPY DISKS OR CASSETTE TAPES, A SUPPLEMENTARY INSTRUCTION MANUAL AND MAY ALSO INCLUDE THE ABOVE MENTIONED BOOK "SOME COMMON BASIC PROGRAMS" WHICH DOES SERVE AS THE USERS MANUAL. THIS BOOK PROVIDES LISTINGS OF ALL OF THE PROGRAMS, EXAMPLE RUNS AND INFORMATION ON HOW TO ALTER THE PROGRAMS IF YOU DESIRE TO CHANGE IT SLIGHTLY FOR A PARTICULAR NEED.

-----

ALPHA-MICRO ALPHA BASIC	/ 1 DISK /	\$15.00 WITHOUT BOOK	/ \$22.50 WITH BOOK
CP/M - CBASIC	/ 2 DISKS/	\$20.00 WITHOUT BOOK	/ \$27.50 WITH BOOK
CP/M - MICROSOFT BASIC	/ 2 DISKS/	\$20.00 WITHOUT BOOK	/ \$27.50 WITH BOOK
TRS-80 LEVEL II BASIC	/ 2 DISKS/	\$20.00 WITHOUT BOOK	/ \$27.50 WITH BOOK
PET BASIC	/ 1 CASS./	\$15.00 WITHOUT BOOK	/ \$27.50 WITH BOOK

ALL ORDERS MUST BE PREPAID AND WE PAY ALL POSTAGE



*THE BASIC BUSINESS SOFTWARE COMPANY, INC.*  
POST OFFICE BOX 2032  
SALT LAKE CITY  
UTAH 84110  
(801) 363-1199

## ELECTRIC MEMO SYSTEM FOR A/M

Dravic Ltd. has developed a simple Data Base Management System for the Alpha Micro which they call the PDQ Electric Memo System. It enables a user to enter information into a data structure called a DATASET and retrieve it without any formal programming knowledge. The simplicity of PDQ lies in the fact that data fields do not have to be set up in advance and information can be entered and accessed upon demand without changing files.

The PDQ System commands are of the simple form:

NOUN' VERB' STUFF or if desired -NOUN'S VERB'S STUFF

Where: NOUN = The subject for which the data is to be stored

VERB = The Sub-subject under which the data is stored

STUFF = The data to be stored

The delimiter ' is necessary to tell the system where the format division is. The system will not work if it is omitted.

To enter phone numbers for example:

NEXT OPERATION? JOHN SMITH' PHONE' 345-6789

or addresses:

NEXT OPERATION? BILL JONES' ADDRESS' 156 ELM ST.

Data is accessed simply by asking for it:

NEXT OPERATION? WHAT'S JOHN SMITH' PHONE or;

NEXT OPERATION? JOHN SMITH' PHONE returns:

JOHN SMITH'S PHONE 345-6789

Asking for a verb such as PHONE will cause the system to return all examples. In this case, all phone numbers.

The use of PDQ Electric Memo System enables a user to start to use the Alpha-Micro System for meaningful work with a minimum of training. Anyone can learn the system in a few minutes. A complete manual is supplied with the system, and there is no limit to the kind of data base operations that can be done with the system, since the files are all accessible by AMOS.

The system retails for \$150.00, Dealer discounts are available.

The PDQ Electric Memo System is being distributed by:

Computer Mart Systems

13 East 30th. St.

New York, N.Y. 10016

Dravic also has an AlphaBasic renumbering program, and a retail inventory program called STORETROL. Contact the Computer Mart for details.

#### UNIVERSAL ENTRY & UPDATE PROGRAM

Micheal's Reality is offering a program that allows operators to create an Entry program through a Specification File. The program supports Defaults, Tables, Globals Skipping, and allows Update, Delete, or Enter transaction types. Prompt sequences may be set to user program requirements.

Object code sells for \$250.00. The source code is also available in structured code with pre-processor for \$500.00.

The pre-processor may be purchased by itself for \$150.00. It supports DOWHILE, DOUNTIL and IF THEN ELSE structures. Source code for the pre-processor is \$250.00. Contact:

Micheal's Reality  
Roy A. Bremer  
16155 W. 12 Mile Rd.  
PO Box 98  
Lathrup Village, Mi. 48076

#### SANTA BARBARA BYTE SHOP SOFTWARE CATALOG

Scott Daley, owner of the Santa Barbara Byte Shop tells us that he has a catalog of business software in BASIC for Alpha Micro available for \$3.98. His programs range from \$250.00 without source code to \$375.00 with source. Programs may also be bought with limited distribution rights for \$500.00. All programs are sent on PERSTD format disks, AMOS version 3.3.

You can order the catalog from:  
Scott Daley  
Santa Barbara Byte Shop  
4 West Mission St.  
Santa Barbara, Ca. 93101

ALPHA MICRO USERS SOCIETY  
c/o COMMUNITY FREE SCHOOL  
PO BOX 1724, BOULDER, COLO. 80306

\*\*\*\*\*  
\* NON-PROFIT ORGANIZATION \*  
\* U.S. POSTAGE PAID \*  
\* PERMIT 490 \*  
\* BOULDER, COLORADO \*  
\* 80306 \*  
\*\*\*\*\*