SOFTWARE

Software is the concept that sets the microprocessor off from other kinds of digital logic. It can make logic adapt to many different applications. It can also become sophisticated enough to allow several micros to talk together harmoniously.

Is software a secret art? For some people it is while others find it quite simple. Some people would rather purchase it and some like to write it. A recent issue of an electronics trade magazine placed the cost of writing one line of code at over $100.

Do you have a piece of software that you would like to see? Submit your ideas to the newsletter and we’ll try to air them. This will allow scrutiny for the readers that are "artists". Once the software is in concrete form it may be aired also. A third step may also be implemented by "purists" who will revise it to conserve memory, save time, or whatever.

There will be a several month turnaround, say don’t expect immediate results. Definitely do not expect $100 a line!!

HARDWARE

The content of most articles I have seen lately is about software. Is there a trend towards purchasing hardware? Perhaps many of the micro users would rather buy than homebrew.

Has anyone given the Aim duties while he is away with answering the telephone? Can the Aim talk or listen yet? Has the Aim learned to walk?

Do you have a need for decals containing the basic keywords from the Short Cut article from the last issue? If there is enough interest perhaps some could be made. No orders are being taken but a rough show of hands will help to see if there is a need.

If you are somewhat unsure how to enter assembly listings a short description has been provided in this issue. There is some interesting reading ahead so I will stop for now.

TTDC
Aim Basic Files

Knut Kvaal
Søndre Skrenten 3 E
1410 Kolbotn, Norway

In the July/August issue, Statistical Analysis, STOK4 read data that was entered via the text editor. An alternate approach to data entry is possible and is explained here.

Aim Data Files requires a machine language program and also some Basic subroutine support. A test program is also provided as an example to illustrate how to integrate the Basic subroutines into a user program.

The machine language routines are essentially the routines presented on pages 7-90 through 7-93 with one major exception— the addition of JSR LL in Close Read File. The example program on page 7-90 should also have this inserted.

The programs assume that Aim format tape is used and that tape 1 will be used for both read and write. To write to tape 2 change the LDA #S00 to LDA#S01 and store

**TEST PROGRAM**

```
10 PRINT"WRIT=1 READ=2" D=2
11 INPUTX
12 IF X=2 THEN 50
13 IF X<>1 THEN 90
20 REM OPEN WRITE
30 GOSUB10200
40 FOR J=0 TO 10
50 PRINT J",SQR(J)

60 NEXT
65 REM CLOSE WRITE
70 GOSUB10300: END
80 REM OPEN READ
90 GOSUB10000
100 FOR J=0 TO 10
110 INPUT X, SQR(J)
111 NEXT
120 REM CLOSE READ
130 GOSUB10130
135 PRINT"RESULT OF READ:"
136 FOR J=0 TO 10
140 PRINT J, SQR(J)
141 NEXT
170 END
```

**SAMPLE RUN**

```
RUN
WRITE=1 READ=2
? 1
ENTER 5 CHR FILENAME
? TEST1
RUN
WRITE=1 READ=2
ENTER 5 CHR FILENAME
? TEST1
RESULT OF READ:
0
1
2 1.41421356
3 1.73205081
4 2
5 2.23606798
6 2.44948974
7 2.64575131
8 2.82842713
9 3
10 3.16227766
```

**GENERAL INFORMATION**

Article contributions are always welcome. Program listings may or may not be retyped. When submitting information on Aim thermal paper adjust the darkness control to its darkest setting. Artwork will not be redrawn so please submit your best work. Artwork will be reduced in size as necessary.

**Back Issues**—Back issues are available starting with Jan/Feb 1979 and later issues at subsequent two month intervals. Back issues are $1.00 in the US and Can, $2.00 elsewhere.

**Time to Renew**—The mailing label contains the last issue that you will receive. If no date appears you have at least two issues left.

The Target—An Aim 65 newsletter is published bimonthly with an annual subscription rate of $5.00 in the US and Can. $12.00 elsewhere (US Funds). First Class and Air Mail respectively. Contact Donald Clem NA/2, Spencerville, OH 43887

THE TARGET

Jan/Feb 1980
MONITOR SUBS

; NOT ALL ARE USED
; RESET
$203F
; CRCK
$3424
; CRLE
$F916
; DUMTA
$56F
; DU11
$5A69
; OUTD13
$1F65
; OUTPR
$7866
; OUTPUT
$897A
; INALL
$8993
; OUTALL
$89BC
; LOAD
$9286
; LOAD1
$9289
; LOADTA
$92F6
; LL
$88FE
; RDUB
$895F
; READ
$893C
; REDOUT
$8973
; TBYTE
$ED5B
; TBYTE
$ED86
; WREGI
$B848
; WREGO
$B871
; PRFLG
$A411
; INFLG
$A412
; OUTFLG
$A413
; NAME
$A42E
; TAPFN
$A434

; TAPOUT
$A435
; DRB
$A800

; TAPEFILE LIST
; VS.1.2
; 28-12-79
; KNUT KAAL
; BARET
; ADDRESS BASIC RETURN

; TBYTE
; AM WITH 4K RAM
; IN7

; POSSIBLE TAPEID
; RECORDER 1 OR 2
; NOT USED HERE
; CONTENT OF NAME
; POKED FROM BASIC
;
; OPEN READ FILE

; EX7
A954 LDA #$T
6D12A4 STA INFLG
A9F0 LDX #$F
A9F0 LDA #$F
; POSSIBLE LDA IN7
6D34A4 STA TAPFN
2D2F63 JSR LOADTA
4CD1CF JMP BARET
; BASIC RET

; FILE

; CLOS READ FILE

; F13 CLOSER
2D2F63 JSR LL
A9CF LDA #$CF
; TURN OFF REC.
2D2F6A8 AND DRB
6D2F6A8 STA DRB
4CD1CF JMP BARET
; RETURN TO BASIC

60 RTS

; OUTPUT-FILE
; OUTS
; CONTENT OF NAME
; POKED FROM BASIC
; POSSIBLE TAPEID
; RECORDER 1 OR 2
; NOT USED HERE
; OPEN WRITE FILE

; EX8
A954 LDA #$T
6D12A4 STA OUTFLG
A9F0 LDA #$F
; REPLACE OUT8
6D34A8 STA TAPEOUT
2D6F65 JSR DUMTA
; RETURN BASIC
4CD1CF JMP BARET

; CLOSE WRITE FILE

; EX9
2D2F69 JSR CRLE
2D2F69 JSR CRLE
2D2F65 JSR DU11
A9CF LDA #$CF
; TURN OFF RECORDER
2D2F6A8 AND DRB
6D2F6A8 STA DRB

; RETURN TO BASIC

END

ERRORS= 0000

10800 REM ENTER FILE ENAME
10810 PRINT "ENTER 5 CHR FILENAME"

10820 INPUTNUS
10821 REM NAME ADR
10850 PS=42309
10840 FOR I=1 TO NUS
10850 ZI=ASC(MIDS(NUS,KI,1))
10860 POKEF,ZI
10870 PS=PS+1:NEXTK
10880 RETURN

READ/WHITE Routines
10000 REM OPEN READ
10010 REM VS.1.3
10020 REM 04/01/80
10030 REM K. KAAL
10050 PR=PEEK(42001)
10060 REM PR=PRINTENAME
10070 REM 6502-PROG
10080 REM START PAGE
10090 FA=15
10100 REM PTR OFF
10110 POKE42001,0
10120 RETURN
10130 REM LOAD
10140 POKE4,19
10150 POKE5,PA
10160 SI=USR(1)
10170 REM RESTORE P D
10180 POKE42001,PR
10190 RETURN
10200 REM OPEN WRITE
10201 PR=PEEK(42001)
10202 REM 6502-PROG
10203 STRT PAGE
10204 PA=15
10210 REM ENTER FILE ENAME
10220 GOSUB10600
10230 REM 04/01/80
10240 POKE4,25
10250 POKE5,PA
10260 SI=USR(1)
10270 REM PTR OFF
10280 POKE42001,0
10290 RETURN
10300 REM CLOSE WRITE
10310 POKE4,51
10315 POKE5,PA
10320 SI=USR(1)
10330 REM RESTORE P D
10340 POKE42001,PR
10350 RETURN

Jan/Feb 1980 THE TAR
ASSEMBLY LISTINGS

This is the first issue that contains assembly language programs. For this reason I will briefly describe how to enter these programs into memory. The Assembler chapter of the Aim User's Guide will also shed some light on this subject.

 AIM Basic Files will be used as an example here. In the listing look for *=SFF00. The "*" represents the location counter and indicates at what portion of memory the program is to reside.

We will start entering the program at $F00 with the memory modify command. We run into a snag right at the beginning. The IN7 is a variable location and is used to store data. If we are going to use the program just as it is then we may just ignore it.

Proceed to *=*+1. This advances the location counter to $3F01. At this point start entering the opcodes in the left six columns. Entering will continue sequentially such as FO1 will contain A9, FO2 contains 54, then FO3 = 8D, etc. When the next *=*+1 is encountered press the space bar to skip over 1 memory location. Then enter data sequentially again. The following is how memory should appear after the program has been entered correctly.

(N)=0F00 XX A9 54 8D ( ) OF24 54 8D 13 A4
OF04 12 A4 A2 00 ( ) OF28 A9 00 8D 35
OF08 A9 00 8D 34 ( ) OF2C A4 20 6F B5
OF0C A4 20 2F E3 ( ) OF30 4C D1 00 20
OF10 4C D1 C0 20 ( ) OF34 FC E9 20 F0
OF14 FE E9 A9 CF ( ) OF38 E9 20 OA B5
OF18 2D 00 A8 8D ( ) OF3C A9 CF 2D 00
OF1C 00 A8 4C D1 ( ) OF40 A8 8D 00 A8
OF20 CO 60 XX A9 ( ) OF44 4C D1 00 XX

Christopher Flynn
2601 Claxton Drive
Herndon, VA 22070

Can any of your readers supply any information on the assembler rom option. It appears to be very similar to the Kim assembler offered by ARESCO. I haven't seen any manuals listed for the assembler.

(Readers without an Aim may want to know that the assembler rom is described in the Aim 65 User's Guide. Can anyone supply a comparison between these assemblers?-Don)
I am an owner of an Aim 65 and subscriber to the Target. I have 4K ram and 8K Basic on my system. I have a little knowledge in basic programming but none in computers and electronics.

1) I want to expand my system to have more memory- 8K or more. Would you explain in detail the parts that I would need and the way I should do it. Would you suggest any parts or boards that need no modification at all so that I can connect it together with ease.

2) I would like to connect my TV to the Aim so that the output would go to the TV instead of the Aim display or printer. How should I do it? Again, parts or boards that require no modification.

3) Would you give an example in basic and assembly language (or machine language) to do a sort so I can learn assembly language. I know how to do it in basic but not in machine language. I can't picture how to do it in machine code.

As a new subscriber to your newsletter, I must say I like it; however, I find most articles too technically oriented. An understanding of the basics is required before they can be of benefit. I wonder if a beginner's section could be started to answer questions such as those above. I am sure there are other Aim owners in the same situation as I am.

---

**1979 Target Index**

<table>
<thead>
<tr>
<th>Software</th>
<th>Basic Short Cut</th>
<th>Nov/Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>Statistical Analysis</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Software</td>
<td>Inside Basic</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Software</td>
<td>Roll</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Software</td>
<td>Mortgage</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Software</td>
<td>Memory Display</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Software</td>
<td>Aim Basic</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Software</td>
<td>For Sale</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Software</td>
<td>Basic Hints</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Software</td>
<td>Symbol Generator</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Software</td>
<td>Enhanced Disassembly to The User VIA</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Software</td>
<td>Enhanced Disassembly Revisited</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Software</td>
<td>Slow Step Routine</td>
<td>Mar/Apr</td>
</tr>
<tr>
<td>Software</td>
<td>Using Existing Software</td>
<td>Mar/Apr</td>
</tr>
<tr>
<td>Software</td>
<td>Aim 65 Quiz</td>
<td>Mar/Apr</td>
</tr>
<tr>
<td>Software</td>
<td>Status Register</td>
<td>Jan/Feb</td>
</tr>
<tr>
<td>Software</td>
<td>A Pseudo Waveform</td>
<td>Jan/Feb</td>
</tr>
<tr>
<td>Hardware</td>
<td>6514 Memory</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Hardware</td>
<td>Keyboard</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Hardware</td>
<td>Best IO Prices</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Hardware</td>
<td>Aim Dimensions</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Hardware</td>
<td>Hardware Products</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Hardware</td>
<td>Sound Generators</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Hardware</td>
<td>I/O Expansion</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Hardware</td>
<td>Aim Physical Connections</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Hardware</td>
<td>Power Supplies</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Hardware</td>
<td>Brain Wave Control</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Announcements</td>
<td>Daim Floppy Disk</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Announcements</td>
<td>Eprom Programmer</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Announcements</td>
<td>Excerpt</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Announcements</td>
<td>Micro Technology Unlimited</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Announcements</td>
<td>Enclosures</td>
<td>Mar/Apr</td>
</tr>
<tr>
<td>Announcements</td>
<td>Software</td>
<td>Mar/Apr</td>
</tr>
<tr>
<td>Announcements</td>
<td>Tiny Basic</td>
<td>Mar/Apr</td>
</tr>
<tr>
<td>Announcements</td>
<td>First Book of Kim</td>
<td>Mar/Apr</td>
</tr>
<tr>
<td>Printer</td>
<td>A Pseudo Waveform</td>
<td>Jan/Feb</td>
</tr>
<tr>
<td>Product Review</td>
<td>Little Buffered Mother</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Product Review</td>
<td>Seawell Memory Board</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Software</td>
<td>Aim 65 Movit</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Display</td>
<td>Display Routines</td>
<td>Jan/Feb</td>
</tr>
</tbody>
</table>

---

Ed Note. A partial answer to the memory expansion question was contained in the July/August issue of the newsletter. A motherboard should be the first selection based on the amount of boards supported by this motherboard or on the supported boards which have the features that you desire. This selection is quite subjective and will not be discussed here. Can any readers supply their ideas on this selection of motherboards.
CHAIN

Steve Bresson
1302 Strawberry Lane
Hanover, MD 21076

Purpose: Controlled loading and execution of multiple files from tape.

Description: Program CHAIN is saved on tape followed by any number of programs that are to be executed. These files must not overwrite CHAIN when loaded. If it is to be executed upon loading, the start address must have been put into the PC storage (using '*' command) and locations A420-A426 saved along with the program. These values are loaded into the 6502 prior to execution. If A420-A426 is not loaded, execution reverts to the monitor.

The user program gives commands to CHAIN by storing data in NAME(A42E).

$00=Quit-Return to monitor
$FF=Ask user for next file name
---(Don't change it)--Load next file
"XXXXX"=File name in A42E-A434, Load that file.

Usage: Load CHAIN. Hit (F1) to start CHAIN. CHAIN takes control from then on.

Locations Used: $080-$FFF: CHAIN
A42E-A434: NAME
A420-A426: G0BK1 (saved regs)

Example: For this example we will use ROLL from the Nov/Dec 1979 issue. To use a program like ROLL, which uses the Editor buffer and pointers:

A) Start the Editor (E) and use the minimum number of locations for the buffer. $280-$480 in this example.
B) Load the text into the Editor.
C) Load ROLL (if it's not in yet) at $617.
D) Make sure it works.
E) Set up saved registers-*=MAIN($840)
F) Now save the necessary memory on tape- $1114:ROLL, pointers, (F1-F3)
$520-$440:Text
$A420-A426:Saved registers

That's it.

Roll is modified from the original so that instead of looping forever, it returns after one pass. If it is to be the last program to be executed, it saves a $00 in NAME before returning.

Sample Run:
0046 BO BCS 0065
0060 A9 LDA #00
0062 8D STA A42E
0065 60 RTS

(remainder of ROLL stays the same, clearing of the display is not performed here.)

(*=0040
(R)
**** FS AA XX YY SS
0040 00 00 00 00 FF
(D)
FROM=A420 TO=A426
OUT=T F=TEST1 T=1
MORE?Y
FROM=0 TO 114
MORE?Y
FROM=200 TO 400
MORE?N

(At this point you may want to enter the Editor and create new text to verify that this program is indeed working)

(D)
FROM=A420 TO A426
OUT=T F=TEST2 T=1
MORE?Y
FROM=0 TO 114
MORE?Y
FROM=200 TO 400
MORE?N

(New text if desired)
0046 BO BCS 0060
(D)
FROM=A420 TO A426
OUT=T F=TEST3 T=1
MORE?Y
FROM=0 TO 114
MORE?Y
FROM=200 TO 400
MORE?N

Execute F80 or set up F1(010C) and use the F1 key. Rewind tape and after prompted enter OUT=T F=TEST1 T=1.

A return to the monitor will be made after three rolls.

THE TARGET

Jan/Feb 1980

Since I LOVE to read letters, I would like readers to submit their choice for the best article or piece of information contained in the 1979 issues. Be sure to vote for yourself because there is a $10.00 prize.


Jan/Feb 1980 THE TARGET 7
Steve Bresson

How many of you realize the capabilities of the Aim 65 basic. The Aim basic saves and loads basic files in ASCII. The TRS-80, Pet and other machines save their programs in non-relocatable hex format. While the Aim method is slower, it has the advantage that a basic file can be edited using the text editor.

The load command does a call of WHERE. This means commands that are executed from the keyboard can be executed from a tape file. The example listing provides a brief look at this technique. If an input line, from anywhere, has a number as its first character, then it is inserted into the program. Otherwise it is executed.

Note. Whenever a line is inserted into a program all variables are lost. So don't try to save data in a variable, insert a line in a program, then try and use that variable again.

?!"BASIC CMD FILE"
?!"TEST OF 12/7/79 SLB"
?!"TAPE INPUT"
DLY=300
FORi=1 TO DLY; i; NEXT
9?!"PROG 1"
10 FOR I=1 TO DLY
20 ?"PROGRAM"
30 ?I
40 NEXT
50 END
RUN
LIST
NEW
5 ?I"PROG 2"
15 FOR I=1 TO 77
25 ?"PROG2 ";I
35 NEXT
45 END
RUN
?!"THE END"
?!"CTRL Z FOLLOWS"
2

(5)
MEMORY SIZE?
WIDTH?
3566 BYTES FREE
AIM 65 BASIC V1.1
LOAD
IN=T F=BASEC T=1
BASIC CMD FILE
TEST OF 12/7/79 SLB
TAPE INPUT
PROG 1
PROG 2
THE END
CTRL Z FOLLOWS
BACK IN BASIC NOW

6502

SOFTWARE--SERVICES--SUPPLIES

Cassettes: High quality, lo-noise, hi-output music grade audio tape wound into 5-screw plastic cases. The perfect solution for taking AIM against data storage cavities.

C-10's: 5 minutes per side.
C-20's 10 minutes per side.

Minimum Order: 20 cassettes

<table>
<thead>
<tr>
<th>20-99</th>
<th>100-199</th>
<th>200+</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-10</td>
<td>$59.95</td>
<td>$49.95</td>
</tr>
<tr>
<td>C-20</td>
<td>$68.95</td>
<td>$54.95</td>
</tr>
</tbody>
</table>

*Your satisfaction guaranteed or return within 30 days for refund.
*Shipped only to USA 48 states
*Pricing includes 2 labels per tape.

Pyramid Data Systems
6 TERRACE AVENUE
NEW EGYPT, NJ 08533

THE TARGET
Jan/Feb 1980
BASIC CHAIN

Steve Bresson

Once you have mastered the concepts in Expose' of Aim Basic and Assembly Language Chain it is time to investigate Basic Chain. Basic Chain uses a little from each of those as an example. The LOAD command uses WHEREI. The subroutines in WHEREI uses the keyboard for input. Since the one desirable characteristic of chain is automatic loading, the keyboard prompting must be eliminated.

If the basic program sets up NAME (42030-42034), INFLG (42002), and TAPIN (42036), it can "LOAD" a tape by setting up the USR( ) function to call LOADTA (58159).

Using Basic Chain

Enter BAS1 and BAS2 into the editor and save each one as a separate file. BAS2 must follow BAS1. Position the tape at the beginning and then do a normal load from basic. The sample run-off shows the proper result.

FROM FILE -BAS1 -

2 NAME=42030
4 INFLG=42002
5 $"="BAS2 "
6 TAPIN=42036
REM USE LDTA INSTEAD
OF LOADTA
8 LDTA=58159
REM TRY TO LOAD ANOTHER
FILE
REM V1A.12.7.79.SLB
9 FOR I=0 TO4:POKE NA
ME+I,ASC(MID$(S$,I+1
,1)):NEXT
11POKE TAPIN,$
13POKE INFLG,ASC("T"
)
15 I=INT(LD+256)
17 POKE $4,LDTA-I*256
6
19 POKE $5,I
21 I=USR($):REM DO IT
1 RUN

FROM FILE -BAS2 -

?!"FILE BAS2"
?!"IT WORKED!!!"
?!"SELL MY CLOTHES,
I'M GOING TO HEAVEN!
"?!"CTRL Z FOLLOWS"
Z

SAMPLE RUNOFF

(5)
MEMORY SIZE?
WIDTH?
3566 BYTES FREE
AIM 65 BASIC V1.1
LOAD
IN=T T=BAS1 T=1
FILE BAS2
IT WORKED!!!
SELL MY CLOTHES, I'M
GOING TO HEAVEN!
CTRL Z FOLLOWS
BACK TO BASIC

Jan/Feb 1980
BOOK REVIEW

Microprocessor Systems Engineering by Camp, Smay and Triska.

This is a good book for the Aim-65 owner who has a minimum of experience with microprocessors. It serves as a basic introduction to microprocessors—pointing out the strengths and tradeoffs involved in their use. Some of the considerations necessary to a good microcomputer design are brought out, with the Aim-65 used as an example.

The book centers around the 6502, with a critical comparison between it, the 6800, and 8080. Each microprocessor’s instruction set, architecture, electrical, and timing is detailed, along with the type of hardware support needed for a minimal system.

The Aim-65 case study delves into the hardware/software of the DISPLAY, PRINTER, KEYBOARD, and TELETYPE interfaces. The software analysis is especially helpful since it lists the pertinent monitor routines, explains how they work, and points out the good features they incorporate.

This book is definitely a good introduction to microprocessors in general, and the Aim-65 in particular.

Steve Bresson

Ed. Note. This book is available from Matrix Publishers, Inc, 30 NW 23rd Place, Portland, OR 97210 for $16.00 (post. paid) or if you are an Aim-65 owner from Marketing Services, RC 55, Rockwell International, PO Box 3669, Anaheim, CA 92803 for $10.00 plus $2.00 handling.

THE TARGET

Jan/Feb 1980