

THE TARGET

January/February 1980

-- an AIM65 newsletter

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

SOFTWARE	AIM BASIC FILES	PAGE 2
SOFTWARE	ASSEMBLY LISTINGS	PAGE 4
SOFTWARE	CHAIN	PAGE 6
SOFTWARE	EXPOSE OF AIM 65 BASIC	PAGE 8
SOFTWARE	BASIC CHAIN	PAGE 9
BOOK REVIEW	MICROPROCESSOR SYSTEMS ENGINEERING	PAGE 10
LETTERS		PAGE 4
INDEX	1979 TARGET INDEX	PAGE 5
ANNOUNCEMENT	RS-232 BY FOBEL	PAGE 9

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 #00 to LDA#01 and store

TEST PROGRAM	SAMPLE RUN
LIST	RUN
9 PRINT"WRITE=1 REA D=2"	WRITE=1 READ=2
10 INPUTX	? 1
11 IFX=2THEN80	ENTER 5 CHR FILENAME
12 IFX<>1THEN9	? TEST1
20 REM OPEN WRITE	RUN
30 GOSUB10200	WRITE=1 READ=2
40 FORJ=0TO10	ENTER 5 CHR FILENAME
50 PRINTJ;" ";SQR(J)	? TEST1
)	RESULT OF READ:
60 NEXT	0 0
65 REM CLOSE WRITE	1 1
70 GOSUB10300:END	2 1.41421356
80 REM OPEN READ	3 1.73205081
90 GOSUB10000	4 2
100 FORJ=0TO10	5 2.23606798
110 INPUTL,SQ(J)	6 2.44948974
111 NEXT	7 2.64575131
120 REM CLOSE READ	8 2.82842713
130 GOSUB10130	9 3
135 PRINT"RESULT OF READ:"	10 3.16227766
136 FORJ=0TO10	
140 PRINTJ;SQ(J):NE	
XT	
170 END	

it at TAPOUT (A435). The Aim user is prompted for the filename.

A taperecorder with remote capability is required. I have a Sankyo ST40 and use a tape gap(A409) of \$20. There is the possibility that either tape will have more data than the program needs or the program will want more than the tape contains. Careful management of file size is a must to avoid this possibility. Remote control and the tape I/O buffers provide the user with unlimited computation time between reading of data.

If you want to move the program and you have an assembler, just change the location counter to the appropriate location. Less changes are needed if one moves the program as whole pages instead of small chunks. The assembly language program may reside anywhere but try to keep it on page boundaries. In this case only the two PA's need to be changed. 10008 and 10203.

PA01 = Open Read
 PA13 = Close Read
 PA23 = Open Write
 PA33 = Close Write

All hex values

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 RR#2, Spencerville, OH 45887

MONITOR SUBS

```

==0000
;NOT ALL ARE USED
==0000 RESET
    =E0BF
==0000 CRCK
    =EA24
==0000 CRLF
    =E9F0
==0000 DUMPTA
    =E56F
==0000 DU11
    =E50A
==0000 OUTDIS
    =EF05
==0000 OURPRI
    =F000
==0000 OUTPUT
    =E97A
==0000 INALL
    =E993
==0000 OUTALL
    =E9BC
==0000 LOAD
    =E2E6
==0000 LOAD1
    =E2E9
==0000 LOADTA
    =E32F
==0000 LL
    =E8FE
==0000 RDRUB
    =E95F
==0000 READ
    =E93C
==0000 REDOUT
    =E973
==0000 TIBYTE
    =ED3B
==0000 TOBYTE
    =F18B
==0000 WHEREI
    =E848
==0000 WHEREO
    =E871
==0000 PRIFLG
    =A411
==0000 INFLG
    =A412
==0000 OUTFLG
    =A413
==0000 NAME
    =A42E
==0000 TAPIN
    =A434

```

```

==0000 TAPOUT
    =A435
==0000 DRB
    =A800

;TAPEFILE LIST
;VS.1.2
;28-12-79
;KNUT KVAAL
;
==0000 BARET
    =C0D1
;ADRESS BASIC RETURN
==0000
    *=F00
==0F00
;AIM WITH 4K RAM
==0F00 IN7
    *=*+1
;POSSIBLE TAPEID
;RECORDER 1 OR 2
;NOT USED HERE
;CONTENT OF NAME
;POKED FROM BASIC
;
;OPEN READ FILE
;
==0F01 EX7
A954 LDA #'T'
8D12A4 STA INFLG
A200 LDX #00
A900 LDA #00
;POSSIBLE LDA IN7
8D34A4 STA TAPIN
202FE3 JSR LOADTA
4CD1C0 JMP BARET
;BASIC RET
==0F13
;
;CLOSE READ FILE
;
==0F13 CLOSER
20FEE8 JSR LL
A9CF LDA #CF
;TURN OFF REC.
2D00A8 AND DRB
8D00A8 STA DRB
4CD1C0 JMP BARET
;RETURN TO
; BASIC
60 RTS

```

```

;OUTPUT-FILE
;
==0F22 OUT8
    *=*+1
;CONTENT OF NAME
;POKED FROM BASIC
;POSSIBLE TAPEID
;RECORDER 1 OR 2
;NOT USED HERE
;
;OPEN WRITE FILE
;
==0F23 EX8
A954 LDA #'T'
8D13A4 STA OUTFLG
A900 LDA #00
;REPLACE OUT8
8D35A4 STA TAPOUT
206FE5 JSR DUMPTA
;RETURN BASIC
4CD1C0 JMP BARET
==0F33
;
;CLOSE WRITE FILE
;
==0F33 CLOSEW
20F0E9 JSR CRLF
20F0E9 JSR CRLF
200AE5 JSR DU11
A9CF LDA #CF
;TURN OFF RECORDERS
2D00A8 AND DRB
8D00A8 STA DRB
==0F44
4CD1C0 JMP BARET
;RETURN TO BASIC
.END
ERRORS= 0000

10800 REM ENTER FIL
ENAME
10810 PRINT"ENTER 5
CHR FILENAME"
;
10820 INPUTNU$
10821 REM NAME ADR
10830 P0=42030
10840 FORKI=1TOLEN(
NU$)
10850 ZI=ASC(MID$(N
U$,KI,1))
10860 POKE0,ZI
10870 P0=P0+1:NEXTK
I
10880 RETURN

```

READ/WRITE
ROUTINES

```

10000 REM OPEN READ
10001 REM VS.1.3
10002 REM 04/01/80
10003 REM K. KVAAL
10005 PR=PEEK(42001
)
10006 REM PR=PRINTE
R STATUS
10007 REM 6502-PROG
ADDRESS START PAGE
10008 PA=15
10010 POKE04,1
10020 POKE5,PA
10030 REM ENTER FIL
ENAME
10040 GOSUB10800
10095 SI=USR(1)
10100 REM PTR OFF
10110 POKE42001,0
10120 RETURN
10130 REM CLOSE REA
D
10140 POKE4,19
10150 POKE5,PA
10160 SI=USR(1)
10170 REM RESTORE P
RINTERSTATUS
10180 POKE42001,PR
10190 RETURN
10200 REM OPEN WRITE
10201 PR=PEEK(42001
)
10202 REM 6502-PROG
START PAGE
10203 PA=15
10210 REM ENTER FIL
ENAME
10220 GOSUB10800
10240 POKE4,35
10250 POKE5,PA
10260 SI=USR(1)
10270 REM PTR OFF
10280 POKE42001,0
10290 RETURN
10300 REM CLOSE WRI
TE
10310 POKE4,51
10315 POKE5,PA
10320 SI=USR(1)
10330 REM RESTORE P
RINTERSTATUS
10340 POKE42001,PR
10350 RETURN

```

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 `*=$F00`. 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 `$F01`. At this point start entering the opcodes in the left six columns. Entering will continue sequentially such as `F01` will contain `A9`, `F02` contains `54`, then `F03 =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.

```
(M)=OF00 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 E5
( ) OF0C A4 20 2F E3 ( ) OF30 4C D1 C0 20
( ) OF10 4C D1 C0 20 ( ) OF34 F0 E9 20 F0
( ) OF14 FE E8 A9 CF ( ) OF38 E9 20 0A E5
( ) OF18 2D 00 A8 8D ( ) OF3C A9 CF 2D 00
( ) OF1C 00 A8 4C D1 ( ) OF40 A8 8D 00 A8
( ) OF20 C0 60 XX A9 ( ) OF44 4C D1 C0 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)

Letters

W.R. Tobler
5032 Birchwood
Santa Barbara, CA 93111

Some things I would like information on are;

- 1) Interfacing to RS-232C devices
- 2) Fortran compiler for 6502
- 3) Basic compiler for the Aim-65
- 4) I am looking for a tape recorder/player which can be rewound under cpu control.
- 5) I would like to hear from anyone who has attempted to attach a facsimile recorder to an Aim (or any other computer).

I found Jim Butterfields INSIDE BASIC very useful.

Steve Silber
5815 Southminster
Houston, TX 77035

A small "Gotcha" appears in the Rockwell monitor. In interfacing a line printer, I wanted to use the User output linkage provided in `OUTALL`. When the user's function is called, the accumulator contains the `OUTFLG`, ASCII, rather than the data to be output. The output data has been pushed on the stack. Therefore a PLA must be executed before jumping to any output routine, but after testing the carry bit since the initialization call does not push anything on the stack. Thus a user linked output routine would look like:

```
UOUT: BCS OUT
      (initialization)
      RTS
OUT:  PLA
      (output processing)
      RTS
```

George Sellers

I just noticed that the `LINR1` and `LINR2` from the July/August issue should have a `(S*S)` in the denominator not just `S*S`.

M.P. Szeto
 181 Willett St. Apt. 102
 Halifax N.S.
 Canada B3M 3C4

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.

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.

1979 TARGET INDEX

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

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: 0F80-0FFF: CHAIN
A42E-A434: NAME
A420-A426: GOBK1 (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:

- Start the Editor (E) and use the minimum number of locations for the buffer. 0200-0400 in this example.
- Load the text into the Editor.
- Load ROLL(if it's not in yet)at \$0-70.
- Make sure it works.
- Set up saved registers-*=MAIN(0040)
- Now save the necessary memory on tape- \$0000-0114:ROLL, pointers, (F1-F3)
\$0200-0400: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 B0 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)

**** PS 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 B0 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.

EXPOSE OF AIM BASIC

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 WHEREI. 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 S
LB"
?!"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 ?!"PROG 2"
15 FOR I=1 TO 77
25 ?"PROG2 "; I
35 NEXT
45 END
RUN
?!"THE END"
?!"CTL Z FOLLOWS"
Z
```

Enter and initialise the editor. Enter this into the buffer from the keyboard.

List buffer to tape with a filename of BASIC.

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

Enter basic and load file BASIC and watch.

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

	20-99	100-199	200+
C-10	59¢ea	49¢ea	46¢ea
C-20	68¢ea	54¢ea	51¢ea

*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

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 S$="BAS2 "
6 TAPIN=42036
REM USE LDTA INSTEAD
  OF LOADTA
8 LDTA=58159
REM TRY TO LOAD ANOT
HER FILE
REM V1A.12.7.79.SLB
9 FOR I=0 TO 4:POKE NA
ME+I,ASC(MID$(S$,I+1
,1)):NEXT
11POKE TAPIN,0
13POKE INFLG,ASC("T"
)
15 I=INT(LDTA/256)
17 POKE 04,LDTA-I*25
6
19 POKE 05,I
21 I=USR(0):REM DOIT
!
RUN
```

FROM FILE -BAS2 -

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

SAMPLE RUNOFF

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

Programmable RS-232 interface for Aim/Kim/Sym Microcomputers. Available from Fobel Enterprises, 552 East El Morado, California 91764. Assembled and tested-\$84.95 while a kit is \$69.95.

This interface allows the user to connect a CRT terminal, printer, modem or any device with an RS-232 interface port directly to the Aim/Kim/Sym computer.

An expansion connector is provided allowing the user to further expand his system.

The interface provides 15 programmable baud rates, programmable word lengths, 1, 1 1/2 and 2 stop bits, odd-even parity generation and detection, serial echo mode, programmable interrupt and status register, half and full duplex operations, parity, framing and overrun error detection as well as a standard DB-25 connector.

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 micro-computer 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
c/o DONALD CLEM
R.R. #2, CONANT RD.
SPENCERVILLE, OHIO 45887