Those readers that ordered the 1979 volume should now have them in hand. If by chance you do not have yours, please write. Copies are available for $6.00. As it turned out this volume required significantly more work than originally anticipated. I hope it reflects the additional effort expended.

I have learned of an AIM 65 users club that is located in Iowa. The club fees are $8.00 a year which includes a subscription to a monthly newsletter. This club occasionally makes volume purchases for its members. Contact Cedar Valley Computer Association PO Box 671 Marion, IA 52302.

One place to purchase the PL/65 ROMs is from Compas Microsystems PO Box 687 Ames, IA 50010. The price is $125. There may be other places to purchase these ROMs. A review of PL/65 will appear in a later issue of the newsletter.

An item I would like to touch on this issue is voice I/O for the AIM 65. There has been little published, to my knowledge, on computer (micro) voice. There are commercially available boards that give a specific computer voice capability, but that doesn't help AIM 65 users. What I would like to see is the accumulation of resources and then we could build from there. If any readers are aware of any resources or have an understanding of the concepts, let me know. Perhaps with the collective efforts of readers we can publish a software and hardware model. It should be built with as little hardware as possible to allow readers with little hardware experience to use the model as well.

Tod Looftbourow's "How To Build A Computer Controlled Robot" has a chapter on voice input to a close relative, KIM-1. I believe this book from Hayden could get us off on the right foot.

Thanks to those that responded to my questions in the May/June issue. Most respondees have about 4K of RAM, either or both of the assembler ROM and BASIC ROMs, use both ROM sets equally, are good to intermediate programmers. Many of the answers suggested peripherals such as additional memory, terminals or video displays, but very little was mentioned about software.

Copyright © 1980 Donald Clem Jr

SOFTWARE Touchtone Dialer 2
SOFTWARE Offset Load 6
HARDWARE EPROM Programmer 8
PRODUCT FORTH 14
Larry Hollibaugh
1708 S.E. Bybee
Portland, OR 97202

I have been experimenting with sound effects on my AIM 65, using the AY-3-8910 Programmable Sound Generator (PSG) from General Instrument. While I was hooking it up to my computer, my wife asked me, "So, what can it do?". After a week of whistles and beeps, phasors and falling bombs, she still wanted to know, "What can it do?". So now it dials the telephone. This seems to have answered her question, even though it only sketches the surface of possibilities for this remarkable device.

For those of you unfamiliar with the AY-3-8910 PSG, it is a Large Scale Integration LSI which can produce a wide variety of complex sounds under software control. It contains three tone generators, a noise generator, three mixers, an envelope generator, amplitude control and three D/A convertors. Also in the chip are two 8-bit I/O ports which are independent of the sound generator. This is a complex and powerful device. A detailed discussion of its internal architecture is beyond the scope and intent of this article. The reader is referred to an excellent article by Steve Ciarcia entitled "Sound Off", in the July, 1979 issue of Byte. Or write to General Instrument Corp., Microelectronics Division, 600 W. John St., Hicksville, NY 11801, and request the AY-3-8910 Data Manual.

The purpose of this article is to describe how I used the PSG to produce the dual-tone frequencies used in Touch-Tone dialing, along with a method of storing and retrieving phone numbers to be dialed.

The numbers are stored in memory using AIM's built-in text editor. As many numbers as memory permits may be stored in the text buffer. They must be stored in the format of (LABEL)=(NUMBER) where (LABEL) is any character string to uniquely identify (NUMBER), with the exception that it may not start with the "#" symbol. (NUMBER) is the actual number to be dialed, and it can include dashes, etc., to make it more readable. The number must be terminated with a space or carr. return, and only one entry per line will be recognized by the dialing program. An equals sign (=) is used to separate (LABEL) and (NUMBER).

Listing 1 is the program to select and dial the numbers stored in the text buffer. The program first prompts the user with a slash (/). The number to be dialed may now be specified in one of three ways:

1. By entering enough characters of (LABEL), starting with the first char., to distinguish it from any that precede it in the text buffer. Type RETURN to dial. The program starts at the top of the text buffer and dials the first number whose label starts with the char.'s entered. If no match is found, ERROR is displayed.

2. By entering the "#" symbol as the first character after the prompt. This is used to dial a number that may not be in the text buffer. The program will re-prompt with "#". Now enter the actual number to be dialed, and type RETURN.

3. By typing RETURN as the first character after the prompt. This will redial the last number dialed.

Actual dialing is accomplished via acoustic coupling (holding the phone's mouthpiece up to the PSG's speaker), or direct connection to the phone lines through a phone company-provided interface device. Be careful about legal restrictions of direct connection.

Figure 1 shows how I have the PSG connected on my AIM. If you choose to connect yours differently, the software will need to be changed accordingly. The subroutines SETUP, ADDR, and SBT1 contain all of the 6522 VIA control instructions.

Of special importance is the frequency of the clock for the PSG. The data tables labeled TBL1 and TBL2 at the end of Listing 1 are for a PSG operating at 1,78977 MHz. To use at a different frequency, the following formula may be used to calculate the necessary values for the data tables:

\[
T_{F10} = \frac{f_{CLOCK}}{16\times T}
\]

Where: \(T_{F10}\) = decimal equivalent of tone period (convert to hex for data tables)
\(f_{CLOCK}\) = input clock frequency
\(T\) = desired tone frequency

2 TARGET Sep/Oct 1980
Figure 2 shows the frequencies used in Touch-Tone dialing, and how they relate to the data tables in Listing 1.

The program is fully commented and should be fairly easy to understand. It makes extensive use of AIM’s editor, and monitor subroutines to conserve memory. The routine NXLINE is a loose copy of UPNO in the monitor listing, and the overall search routine is a modified version of AIM’s PCH sub. The tones are generated by the routine starting at TONES.

I hope this helps you explain those strange sounds to your wife. “Look, dear, it does something!”.

<table>
<thead>
<tr>
<th>TELEPHONE DIGIT</th>
<th>FREQUENCIES (Hz)</th>
<th>HEXDECIMAL DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>941,1336</td>
<td>77,54</td>
</tr>
<tr>
<td>1</td>
<td>697,1209</td>
<td>40,5C</td>
</tr>
<tr>
<td>2</td>
<td>697,1336</td>
<td>40,54</td>
</tr>
<tr>
<td>3</td>
<td>697,1477</td>
<td>40,4C</td>
</tr>
<tr>
<td>4</td>
<td>770,1209</td>
<td>91,5C</td>
</tr>
<tr>
<td>5</td>
<td>770,1336</td>
<td>91,54</td>
</tr>
<tr>
<td>6</td>
<td>770,1477</td>
<td>91,4C</td>
</tr>
<tr>
<td>7</td>
<td>852,1209</td>
<td>83,5C</td>
</tr>
<tr>
<td>8</td>
<td>852,1336</td>
<td>83,54</td>
</tr>
<tr>
<td>9</td>
<td>852,1477</td>
<td>83,4C</td>
</tr>
<tr>
<td>#</td>
<td>941,1207</td>
<td>77,5C</td>
</tr>
<tr>
<td></td>
<td>941,1477</td>
<td>77,4C</td>
</tr>
</tbody>
</table>
TOUCH-TONE TELEPHONE DIALER
; USING THE AY-5-8910 PSG

; BY LARRY R. HOLLIBAUGH

; ZERO PAGE USAGE:
NOWLN=$DF
STRING=$BB
FLAG=$FE
YSAV=$FF

; 6522 VIA ACCESS:
UIRB=$A000
UIRA=$A001
UDRB=$A002
UDRA=$A003
;MONITOR EQUATES
CKERO=$533E
EQUAL=$5736
PLS1=$5837
BLANK=$5838
READ=$593C
RDRUB=$235F
OUTPUT=$597A
CRCK=$5A24
HEX=$5A7D
PLNC=$F727
TOPNO=$F83C
ATBNT=$F899
ADDA=$F92A

; USER ACCESS HOOK;
*=$10C
JMP DIAL

*=$200

; MAIN PROGRAM STARTS HERE
ENTRY POINT TO DIAL USER SPECIFIED #
DIAL JSR PSL1; PROMPT USER WITH */*
LDY #0; SET UP FOR DELETES
FIND1 JSR RDRUB; INPUT CHARACTER
CPY #0; FIRST CHAR?
BNE FIND2
STY FLAG; INIT FLAG FOR TEST SRCH
CMP #$D; REDIAL LAST #?
BEQ REDIAL
CMP ""; CODE FOR NUMERICAL ENTRY
BNE FIND2
STA STRING
JSR EQUAL; PROMPT USER WITH "=
INC FLAG; BYPASS TEXT SRCH
INY; ACCOUNT FOR "=" IN STRING
FIND2 CMP #$D; DONE?
BEQ FIND3
STA STRING,Y; SAVE IT
INY; NEXT CHAR

4 TARGET Sep/Oct 1980

; CPY #$10; 16 CHAR'S MAX
BNE FIND1
FIND5 LDA #$D; IN CASE OF NUM. ENTRY:
STA STRING,Y; END STRING WITH CR...
LDA #$STRING; SET UP STRING...
STA NOWLN; AS CURRENT LINE
LDA #$STRING
STA NOWLN+1
LDA FLAG; FLAG NOT ZERO MEANS...
BNE REDIAL; NUMERIC ENTRY-DIAL IT
;INITIATE TEXT SEARCH
STY YSAV; CHARACTER ENTRY COUNT
JSR TOPNO; SET NOWLN-TOP OF BUFFER
JMP FIND5; START SEARCH
NXLINE LDY #0
JSR ATBNT; CURRENT LINE AT BOTTOM?
BCS NONE
NXT1 LDA (NOWLN)Y; TEST FOR END
BEQ NONE; OF TEXT BUFFER
INY
CMP #$D; LOCATE END OF LINE
BNE NXT1
TYA
JSR ADDA; ADVANCE TO NEXT LINE
FIND5 LDY #0; START WITH FIRST CHAR
FIND7 LDA (NOWLN)Y; GET CHARACTER
BNE FIND6; END OF TEXT BUFFER?
NONE JMP CKERO; END OF BUFFER=NO MATCH
FIND8 CMP #$D; END OF LINE?
BEQ NXLINE
CMP STRNG,Y; MATCH?
BNE NXLINE; NO-NEXT LINE
INY; YEQ-NEXT CHAR
CPY YSAV; ALL CHAR'S TESTED?
BNE FIND7
;
REDIAL JSR CRCK; PRINT DISPLAY
JSR BLANK; OUTPUT A SPACE
JSR PLNC; SHOW CURRENT LINE
JSR SETUP; GUARANTEE 6522 STATE
JSR CLRPSG; RESET PSG
LDY #0
TAB INY; ADVANCE THROUGH LINE...
LDA (NOWLN)Y
CMP #$D; UNTIL CR...
BEQ EXIT
CMP "="; OR EQUAL SIGN FOUND
BNE TAB
TONES INY
LDA (NOWLN)Y; NEXT # TO BE DIALED...
CMP #$D; OR "SPACE" ENCOUNTERED
BEQ EXIT
JSR HEX; CONVERT TO HEX
BCS TONES; IGNORE NON-NUMERIC
STY YSAV; SAVE # POINTER
TAY; USE Y AS POINTER TO PSG DATA
LDX #0 ; PSC ADDR
LDA THL1,Y ; VALUE FOR PSC TONE A
JSR SRT1 ; SEND TO PSC
LDX #2 ; PSC ADDR
LDA TBL2,Y ; VALUE FOR PSC TONE B
JSR SRT1 ; SEND TO PSC
LDX #7 ; PSC ENABLE REG ADDR
LDA #2F ; ENABLE TONES ON CH A&B
JSR SRT1
INX ; PSC CH A AMPLITUDE
LDA #8F ; MAX AMPLITUDE
JSR SRT1
INX ; PSC CH B AMPLITUDE
JSR SRT1 ; MAX AMPLITUDE
LDX #3C ; VALUES FOR TONE DURATION
LDY #0
JSR DELAY
JSR CLRPSG ; TURN OFF TONES
LDX #14 ; VALUE FOR SILENCE DURATION
JSR DELAY
LDY YSAV ; RECOVER # POINTER
JMP TONES ; DIAL NEXT NUMBER
; SUBROUTINES START HERE
SETUP LDS UDBR
ORA #3 ; DDBR BITS 0 & 1 = OUTPUTS...
STA UDBR3 ; BITS 2 - 7 UNCHANGED
LDA UDBR
AND #7F ; DDBR BITS 0 & 1 = 00...
STA UDBR2 ; BITS 2 - 7 UNCHANGED
LDA #7F
STA UDBRA ; DDRA = OUTPUTS
EXIT RTS
;
CLRPSG LDX #3F ; PSC PSG ADDR
LDA #0 ; PSG DATA
CLRP JSR SRT1 ; WRITE TO PSG
DEX ; NEXT ADDR
BPL CLRLP ; UNTIL ALL SET TO ZERO
RTS
;
ADDRS STX UDBR ; X = ADDR TO PSG
PHA
LDA UDBR3
ORA #3 ; "LATCH ADDRESS" MODE
ADDRS STA UDBR
AND #7F ; "PSG INACTIVE" MODE
STA UDBR
PLA
RTS
;
SRT1 JSR ADDR ; WRITE X TO PSG AS ADDR
STA UDBR ; ACC TO PSG AS DATA
PHA
LDA UDBR3
ORA #2 ; "WRITE TO PSG" MODE
BNE ADDR1 ; BRANCH ALWAYS
;
DELAY DEX
BNE DLYP
RTS
DLYP DEX
BRQ DELAY
JMP DLYP
;
; DATA FOR PSG TONE FREQUENCIES
; ONE BYTE FROM EACH TABLE FOR
; DUAL TONES. C-F = SILENCE
TBL1 .BYT $77,$70 ; 0,1
.BYT $A0,$A0,$91,$91 ; 2,3,4,5
.BYT $91,$83,$83,$83 ; 6,7,8,9
.BYT $77,$77,0,0,0,0,0,0 ; *A,C,D,E,F
;(*A,#B)
;
TBL2 .BYT $54,$5C ; 0,1
.BYT $54,$4C,$5C,$54 ; 2,3,4,5
.BYT $4C,$5C,$54,$4C ; 6,7,8,9
.BYT $5C,$4C,0,0,0,0,0,0 ; *A,#C,D,E,F
;(*A,#B)
;
LAST .END

;AV-3-8910 "SOUND"
;GEN PURPOSE SUBS
;BY LARRY HOLLIBROUGH
;ZERO PAGE USAGE:
;PARAM ****10
;XSRV ****1
;6522 VIA ACCESS:
;=9000
;UDBR ****1
;UDBR ****1
;UDBR ****1
;=9020
;INITIALIZE 6522 VIA SETUP LDS #3
;STA UDBR
;SETUP1 LDS #0
;STA UDBR
;LDA #FF
;STA UDBR
;RTS
;READ FROM PSG AT
;ADDR REF. BY X
;FETCH JSR ADDR
;RDPSC LDS #0
;STA UDBR
;LDA #1
;STA UDBR
;LDA UDBR
;PHA
;JSR SETUP1
;PLA
;AND BITS X
;RTS
;BITS EVT #F #F
;BVT #F #F #F #F
;BVT #F #F #F #F
;BVT #F #F #F #F
;BVT #F #F #F
;LST END
Steve Bresson
1302 Strawberry Lane
Hanover, MD 21076

OFFSET LOAD

This program loads an object program from tape at an offset from the save address. This is useful for those who have PROM programmers. It could also be useful for relocatable programs - save one copy to tape, then load it into where it's needed.

Assemble the program to tape, load at an offset so the program is in RAM, then program the PROM with it. It displays the number of hex bytes in each record, the save address, and one past the end address.

```
0000 ADONE=$EA81
0000 ADDR=$41C
0000 CRLOW=$EA13
0000 WHERE1=$E848
0000 INALL=$E933
0000 CLK=<$E4B
0000 CHEKAR=$E548
0000 NUMA=$EA46
0000 BLANK2=$E838
0000 WRITAZ=$E208
0000 OUTPUT=$E97A
0000 RBYTE=$E3F0
0000 STBYTE=$E412
0000 CKSUM=$A41E
```

---END---
POWERFUL AIM-65 FLOPPY DISK SYSTEM

Everything is here. Just plug it in and start execution at $9000, then ADOS Operating System is running.

System includes:
- FP-950 Controller Board.
- ADOS Operating System on a 2532 EPROM.
- Single sided/double density drive with power supply & case. One Dysan diskette.

* Powerful ADOS Operating System allows access of Basic data as well as program files IN and OUT from floppy disk under AIM-65 Basic program control (available library utility).
* Able to assemble multiple source files from disk and routing object code or listing outputs directly to either: disk, memory or to the on board compatible centronics printer port.
* Able to save and load text/object files directly from the Editor or Monitor on the AIM-65.
* Includes information on accessing the disk files from user program control.
* Able to execute programs directly from disk.
* Dynamic allocation of files.
* Block transfer mode allows loading 32 K bytes in 4 seconds.


---

DSAIM-65

The DSAIM-65 is a really complete low cost development system for the 6500 microprocessor family with unsurpassed capability for just

PRICE $3795

The system includes:
- 1 AIM-65 Microcomputer.
- 1 High level language BASIC ROM set.
- 1 FP-950 Mini-Floppy Disk Controller module, capable of driving up to 4 single/double sided, single/double density drives. With Centronic's type printer interface.
- 1 ADOS Operating System on 2532 EPROM (see powerful AIM-65 Floppy Disk System above)
- 1 Dysan diskette containing library utilities to support the handling of data files under BASIC language program control.
- 2 Single sided, double density drivers with power supply and metal cases.
- 2 Interface cables for floppy drivers
- 1 CRT-80 Video Controller module with 2k bytes of software on board to replace the 20 character single line display for 80,64 or 40 characters by 25 lines video display. Upper/lower case ASCII characters, 138 semigraphic characters plus 18 control characters. Refresh RAM not on address space of CPU with no wait states when updating. Reverse video for each character.
- 1 Video Monitor for 80 x 25 , 64 x 25 or 40 x 25 lines.
- 1 Coax cable for video monitor.
- 1 MB-32K Dynamic RAM module, capable to extend the memory up to 64 K bytes. Single +5 v power supply. Totally transparent refresh, no cycle stealing. Memory selectable in 4K blocks by switches. Allows bank selection.
- 1 EMB-6 Expansion Mother Board. Enabled in 4K increments by switches. Fully buffered. 6 card slots. EXORciser bus compatible. Straight extended from expansion connector on the AIM-65
- 1 Power supply with +5V @ 6A , 24 V @ 1A
- 1 730 Centronics dot matrix printer

Without printer substrate $845

APPLIED BUSINESS COMPUTER CO.
707 S. STATE COLLEGE, SUITE G
FULLERTON, CA. 92631 TEL.(714) 871-1411
NEW PRODUCT

A version of the FORTH programming system is now available for the 6502 based KIM-1, SYM-1, and ATM 65 microcomputer systems. A version for the APPLE II will be forthcoming.

FORTH is a high-level vocabulary based language which is structured and extendable. It's finding its way to increased usage in microcomputer applications such as graphics, robotics and process control and telecommunications due to its memory efficiency, fast operating speed (when compared to most other high-level languages) and ease of interfacing to assembly language.

But, perhaps the greatest advantage of using FORTH is that every time you program a new application language, a new application language is created because FORTH actually becomes the application! A typical FORTH application language/program could consist of words like DELAY, READ-A/D, TESTVALVE, TURNON and SCREENCLEAR.

Non-programming engineers can be taught to write programs in this new applications language in less time than it would take to teach them a conventional high-level language, like BASIC.

This particular version of FORTH contains a built-in 6502 assembler (this enables the user to "drop-in" to assembly language at any time), a text editor (to manipulate FORTH source programs), and a cassette file management system. Information on interfacing FORTH to a floppy disk is provided as well as several extensions to the language.

6502 FORTH sells for $90.00 (plus $4.00 for 32K) and includes a user manual, a well-commented source listing, and a cassette containing the object code.

For ordering or additional information, contact: Eric C. Rehnke, Tech Services, 1067 Jadestone Lane, Corona CA 91720.
EPROM PART NUMBER
Start Program here
2708
PROGRAM START ADDRESS
=0200
EPROM ADDRESS
FROM=000 TO=00F
CHK IF BLANK? Y/N
READ
PROGRAM? Y/N
PROGRAM 0001020304050607080910111213141516171819202122232425262728293031323334
5363738394041424344454647484950515253545556575859606162636465666768697071727374
5767778798081828384858687888990919293949596979899
VERIFY? Y/N
VERIFY
VERIFY COMPLETE

End program

Data that was programmed

Re-entered program

Data in EPROM

These addresses were not programmed

SEP/OCT '80 TARGET 15
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 may be oversize if necessary and will be reduced to proper size.

Text should accompany articles to explain what is being done, how it is done, and how it may be modified to suit the user.

Please submit a self addressed stamped envelope for any replies that you desire.

Back Issues—A consolidated 1979 issue is available for $6.00 ($12). In addition 1980 issues are available beginning with the January/February and at subsequent two month intervals. Individual 1980 issues are $1.00 (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 on you subscription.

Target—An AIM 65 newsletter is published bimonthly with a subscription rate of $6.00 in the US and Canada, $12.00 elsewhere. Contact Donald Clem, RR #2, Spencerville, OH 45887.

THE TARGET

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

14 TARGET SEP/OCT 1980