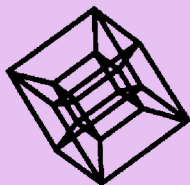


APPLE II® COMPATIBLE

APPLE MUSIC II



A L F PRODUCTS

The
Apple Music II
Owner's Manual

Complete Instructions
for the
10-5-1
Apple Music II

the Dancing Bear
is otherwise engaged
and thus
cannot appear in this manual

Copyright © 1980
ALF Products Inc.
1448 Estes
Denver, CO 80215
U.S.A.

Part Number 11-1-1A

The information contained in this manual was believed to be accurate at the time of publication. Although this manual has been carefully checked for accuracy by our inebriated technical staff, we assume no responsibility for errors or omissions. ALF reserves the right to make changes in the products and/or specifications without notice.

this manual is dedicated to
the Cybernetic Empire

Project Engineer: John Ridges
Software Design: Tim Gill, John Ridges
Manual written by: Philip Tubb
Manual graphics by: Rick Harman
Index by: Tim Gill, Forrest Thiessen
Prior work: 10-5-16 crew

Comments and suggestions can be sent to the address on the preceeding page.

"Apple][" is a trademark of Apple Computer Inc.

CONTENTS

1. INSTALLATION

- 1-2 Operating tips
- 1-3 Problem checklist
- 1-3 Disk software
- 1-3 Backup

2. ENTRY

- 2-1 Entering a simple song
- 2-12 Correcting mistakes
- 2-15 Entering rests
- 2-16 Subroutines
- 2-19 Loading and saving songs
- 2-20 Adjusting the tempo
- 2-21 Envelopes
- 2-27 Recommended reading
- 2-27 Beaming
- 2-28 Sample song breakdown
- 2-29 Summary of commands
- 2-29 Type 1 commands
 - 2-29 Note duration
 - 2-29 . 3
 - 2-30 # b ♯
 - 2-30 → ←
 - 2-30 INS
 - 2-30 †
 - 2-30 GOTO
 - 2-31 INTEGER
 - 2-31 LENGTH
 - 2-31 MEASURE
 - 2-31 PART
 - 2-31 PLAY
 - 2-32 SAVE
 - 2-32 ***DISK
- 2-32 Type 2 commands
 - 2-32 DEL
 - 2-32 DELETE
 - 2-33 EDIT
 - 2-33 LOAD
 - 2-33 NEW
 - 2-34 SPEED
 - 2-34 STEREO
 - 2-34 SUBROUTINE
- 2-35 Type 3 commands
 - 2-35 KEY
 - 2-35 QUARTER
 - 2-36 TIME
- 2-36 Type 4 commands
 - 2-36 REST
 - 2-36 Paddle 1
 - 2-37 TIE
 - 2-37 ATTACK
 - 2-37 CALL
 - 2-37 DECAY
 - 2-38 FUZZ
 - 2-38 GAP
 - 2-38 POKE
 - 2-38 RELEASE
 - 2-38 SUSTAIN
 - 2-39 TEMPO
 - 2-39 TRANSPOSE
 - 2-39 VOLUME
- 2-39 Tips
 - 2-39 Partial starting measure
 - 2-39 Rests at the end of Parts
 - 2-40 Paddle settings
 - 2-40 "Backup"
 - 2-40 TRANSPOSE
 - 2-40 Copying songs without ENTRY

- 2-40 Reset
- 2-40 Integer/Applesoft switch
- 2-41 Song data format
- 2-41 Selected hex addresses

3. PLAY

- 3-1 LOAD
- 3-1 PLAY
- 3-1 STOP
- 3-2 INT or FP

4. DISCO

- 4-1 To add songs
- 4-2 To start over
- 4-2 Using "START" and "END"
- 4-2 Using more than one disk drive

5. PROGRAMMING WITH PERFORM

- 5-2 An example
- 5-3 A few precautions
- 5-6 Synthesizer initialization
- 5-6 Song configuration
- 5-7 Reading the "suggested speed"
- 5-7 Tempo control
- 5-7 A sample session
- 5-9 Technical
- 5-9 Type A commands
 - 5-9 CHANNEL NUMBER
 - 5-9 CALL
 - 5-10 RETURN
 - 5-10 STOP
 - 5-10 END
- 5-10 Type B commands
 - 5-10 TRANSPOSE
 - 5-11 FUZZ
 - 5-11 GAP SIZE
 - 5-11 ATTACK RATE, DECAY RATE, VOLUME LEVEL, SUSTAIN LEVEL, RELEASE RATE
- 5-11 Type C commands
 - 5-11 PITCH
 - 5-12 REST
- 5-12 Song data
 - 5-12 Relative addresses
 - 5-12 Start of data
 - 5-12 Part data
 - 5-13 Subroutine data
- 5-13 TEMPO command
- 5-13 Temporaries
- 5-14 Command numbers
- 5-15 Block diagram

6. PROGRAMMING BARE HANDED

- 6-1 Initialization
- 6-1 Volume
- 6-1 Frequency
- 6-2 6502 programming
- 6-2 Divisor calculation

7. TECHNICAL

- 7-1 PERFORM listing
- 7-9 Circuit board photo
- 7-9 Schematic
- 7-11 Repair diagram

This manual is published in a three-hole punched format, and is furnished with a reusable binder. Three-hole punched manuals have long been standard in the computer industry due to their versatility. They are easy to issue correction pages and addendums for, and the user can combine several manuals on similar topics into a single binder. ALF Products is proud to continue this fine tradition, and we hope you will enjoy this format.

1

INSTALLATION

THIS MANUAL DOES NOT COVER USE OF THE APPLE II COMPUTER. READ THE MANUALS SUPPLIED WITH YOUR APPLE, AND FAMILIARIZE YOURSELF WITH ITS USE, BEFORE CONTINUING.

PLEASE READ THIS ENTIRE SECTION BEFORE BEGINNING.

Installation of your Apple II compatible music synthesizer is easy. Just follow these instructions:

1. You will need an audio amplifier and speakers or a home hi-fi system. Turn your amplifier off and the volume all the way down.
2. Turn the Apple off and remove the top cover.
3. Attach the audio output cable to the synthesizer. You'll notice that the connector on the end of the audio cable can be plugged into the 3-prong connector on the synthesizer circuit card in either of two ways: with the slots in the plastic housing toward the circuit card or away from it. You may plug it in either way. Just be sure all three prongs go into the three holes in the plastic connector. Reversing the connector will reverse the stereo output (Left becomes Right and vice-versa.)
4. Plug the synthesizer into an expansion slot. Any slot may be used. Route the cable out through one of the holes in the back of the Apple. Replace the top cover of the Apple.
5. Plug the audio cable into your amplifier or home hi-fi system. Any of a variety of inputs may be used, such as Aux (or Auxiliary), Tuner, or Tape Play. Do not use Phono (phonograph) inputs. Connect one plug to the Left input and the other to the Right input of the same type (e.g. Aux left and Aux right). When using the synthesizer, set the amplifier to select the input used (Aux or Tuner, or Tape for "tape play" or "tape in"). Note: if a mono amplifier is to be used, a "Y-adapter" can be purchased from your local stereo store, or you can order the 1Ø-1-2 Mono Cable (which has only one RCA-type phono plug, and mixes the Left and Right signals together) from your dealer or from ALF.
6. The synthesizer is supplied with several programs, on cassette tape or on disk. Normally the programs written to run using Integer BASIC should be used. If you do not have Integer BASIC, use the Firmware Applesoft versions. In this manual, these will be referred to as the Applesoft versions although they will not work with the version of Applesoft supplied on cassette tape for use with Integer BASIC Apples. Note that when using Applesoft, FP must be

typed anywhere this manual says to type INT.) Each program which uses the synthesizer has a line which contains information regarding the slot number of your synthesizer. This line is always located at line 10. As supplied, all programs are for use with the synthesizer plugged into slot 4. If you are using a different slot, you will need to change some of the programs. Each program must be loaded, line 10 modified, and then saved. At the beginning of the instructions for each program in this manual the exact procedure required is explained.

IMPORTANT: When changing line 10 you must load the program, change line 10 carefully making sure the length of the line is not changed, and then save the program. You must not save a program after it has been run, since it has then modified itself and thus will not contain many important statements which were originally present.

7. Turn your amplifier on. You are now ready to use the synthesizer. Adjust the volume on your amplifier once you begin playing songs.

OPERATING TIPS

Plug your Apple and amplifier into the same electrical outlet if possible. Differences in ground potentials can cause difficulties when different outlets are used. If different outlets must be used, or if the amplifier does not have a three-prong (grounded) power cord, do this: when removing the synthesizer from the Apple, always unplug the audio cable from the amplifier first. Similarly, plug the synthesizer into the Apple prior to connecting the audio cable into the amplifier.

Always turn the Apple off before inserting or removing any circuit card.

Any Apple circuit board can be damaged by excessive static. This particular circuit board has been carefully designed to minimize the possibility of damage (since only TTL type chips are used). However, walking across a carpet while holding an Apple circuit card can "charge" you and the card to voltages high enough to damage any electronic circuit. Therefore, you should always hold the circuit card in one hand, and touch the metal case of the Apple power supply with the other hand prior to inserting a board in the Apple. This will allow the static charge to be drained through the third prong (ground prong) of the power cord, rather than through the circuit card and the Apple circuits.

Should your synthesizer ever need repair, return the entire unit (including the audio cable and software) to your dealer or to ALF. Be sure to include a description of the problem. Your dealer can repair the synthesizer if he is an

ALF-authorized service agent; otherwise he can return it to our factory service department for prompt attention. Replacement parts, such as a new audio cable or owner's manual, can be obtained from your dealer or from the factory.

PROBLEM CHECKLIST

1. Load the program you are using. List line 10. Is it correct? If not, refer to the instructions for the particular program being used.
2. If no sound is produced, check the audio cable connections. Make sure all 3 pins go into the plastic connector.
3. Check connections to the amplifier and all switch settings on the amplifier. Do the amplifier and speakers work with other sound sources? If not, replace.

DISK SOFTWARE

When using the disk version of the software, you should "boot" the disk (see your Apple Disk II manual). The HELLO program which then runs will ask you which slot your synthesizer is in. It will then automatically configure the ENTRY and PLAY programs, so it will not be necessary for you to change line 10 in either program. If you ever change which slot your synthesizer is in, run the HELLO program to reconfigure ENTRY and PLAY.

BACKUP

ALF authorizes the original retail purchaser of this synthesizer to make one copy of each disk or tape supplied with the synthesizer (on any medium) for backup use only. Each backup copy should be kept in a safe place so it can be used to recreate the original copy should it ever be erased. Each copy may be as supplied, or with any modifications the user may require, and must bear the notice:

Copyright (P) 1980 by ALF Products Inc.

Other reproduction of these recordings on any media; in data, audio, or any other form; with or without modifications; is prohibited by federal law and is subject to criminal prosecution.

2

ENTRY

The ENTRY program is used to enter and play songs. Notes, rests, and other musical parameters are entered in a convenient sheet-music type format displayed on the screen (video monitor), and selected from a "menu" of available notes which is also shown on the screen. Songs entered can be stored on (and loaded from) cassette tape or disk. A variety of other functions are available for editing, stereo selection, and so forth.

To run this program, you must have 24K or more memory. If you are using a DISK II, you need 32K or more. Very detailed graphics are presented on the screen, so it is recommended that a black and white monitor (such as the Sanyo VM4209 or VM4215) be used rather than a television set, although good results have been obtained using the Sup'r'mod II UHF channel 33 TV interface unit (from M&R Enterprises) and the Sony Trinitron model KV 1513 color television.

You must also have the Apple's paddles plugged in. All programs which use the synthesizer require the paddles.

First, load the program from disk or cassette tape. List line 10. It will be 10 SLOT=4. Carefully retype the line changing only the digit 4 to the proper digit for your system. Now save the program on your disk. If you do not have a DISK II, save the program using your own recorder to improve loadability. The program is now configured for your system, and can be run any time you like without having to change line 10. If you ever change the slot position of your synthesizer, you should do this configuration procedure again.

ENTERING A SIMPLE SONG

Load the program if it is not currently in memory. Type RUN and press return. The screen will go to hi-res graphics mode and display:



The number in front of "FREE" will vary according to memory size and other factors. It indicates the number of notes which can be added, and will be constantly updated as you enter and edit the song.

The first six measures of "America" are shown below:



In order to enter the piece using ENTRY, it is first necessary to break the piece up into "parts". Each part is an independent melodic line in which at most one note is played at a time. It is best to choose each part so it is consistently from the same melodic line in the music. This allows you to select appropriate envelope settings for each line later on. The first part, called Part 0, is shown below. It is the main melody.

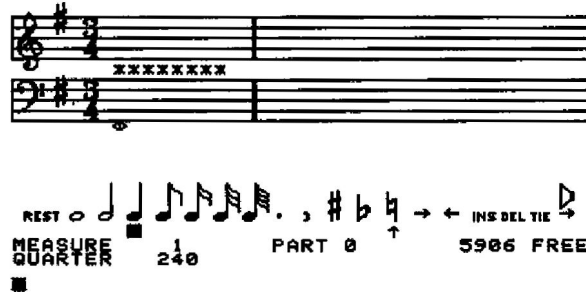


To begin entering a new song, type NEW and press return. ENTRY will display "NUMBER OF PARTS?". Just press return. This will make the song have only 1 part (part 0). ENTRY now displays "SUGGESTED SPEED?". Since we don't really know what the playback speed should be yet, just press return. ENTRY will assume a speed of 255 (the slowest speed). ENTRY now displays "TITLE LINE 1". If you wish, you can type in a line which will be shown on the screen when the song plays. If you're not in the mood, just press return. The title lines can always be entered (or changed) later. ENTRY will then ask for title lines 2 through 4. Type titles if you like, or just press return for each line.

Part 0 can now be entered. Note that under "MEASURE 1" the screen shows "KEY C". If you turn paddle 1's knob, a small flying saucer will move up and down to the left of the two 4/4's. (If you get paddle 0 by accident, then a small arrow will move left and right instead. This doesn't matter. Try again with the other knob.) This flying saucer is called the "cursor", and it is very important. The cursor is a "pointer" to a particular item in the song. Currently, it is pointing to the KEY C before the 4/4. The key of C is a "neutral" key having no sharps or flats, and thus shows only as a blank space right before the 4/4.

Type KEY:1S and press return. A sharp sign will appear before the 4/4, and the cursor will move over to the 4/4. KEY:1S directs ENTRY to write a key signature of 1 sharp (S means "sharp", and F would be used for "flat"). This key signature is written over whatever item the cursor is on. Since it was on the KEY C, the KEY C is overwritten with a KEY 1S.

When the KEY 1S is written, the cursor moves on to the next item in the song, which is a time signature of 4/4. The place on the screen which used to show KEY C now shows TIME 4/4 since the cursor is over the 4/4. "America" has a time signature of 3/4, so type TIME:3/4 and press return. The 4/4 will promptly change to 3/4, and the cursor will move on to the next item. The screen now looks like this:

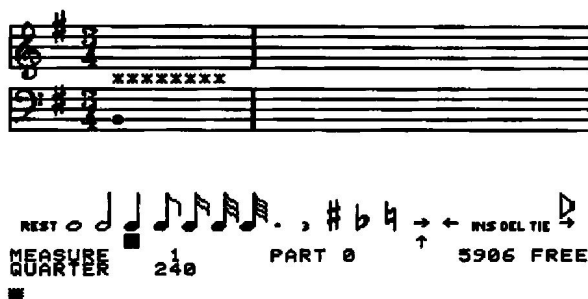


You've only been at this for a few seconds, and already you've told ENTRY two very important facts about "America", the song you're entering. Without these details it would be very difficult to enter the song properly. Why, you're probably half way to being a professional musician, if you weren't one when you started.

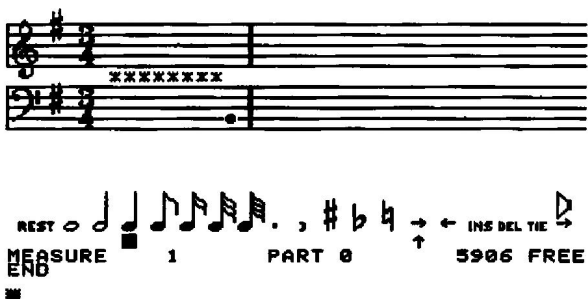
Now the cursor is at the first of eight asterisks (*) displayed between the treble and bass staves, and the item the cursor is at is a QUARTER 240. These eight items are special goodies that describe things about the song which don't display well in sheet music format. This particular one indicates how long a quarter note should play (240 time units per quarter note). While you will eventually want to learn about these, they are not important now, and it is best to skip over them at present. This is done using one of the paddles.

Turn one of the paddle knobs back and forth. If the arrow above "MEASURE 1 PART 0 5906 FREE" moves left and right, you're turning paddle 0, the "menu paddle". If the flying saucer cursor moves up and down, you're turning paddle 1, the "note paddle". Place the menu paddle (paddle 0) on your left and the note paddle (paddle 1) on your right. Usually you'll have your left hand on the menu paddle and your right hand on the note paddle; sometimes you'll have to let go of the paddles to type on the keyboard (probably not very often). Turning a paddle knob with one hand is almost always followed by pressing a paddle button with the same hand. You see, turning the knob selects something (a menu item when turning the menu paddle, or a note position when turning the note paddle), and then pressing the button tells ENTRY to look at the position of the knob and do whatever it is set for. Since the two paddles are used for different purposes, you always press the button of the knob you have just adjusted in order to activate the function you adjusted the knob to indicate.

For example, using your left hand only, position the menu paddle so that the upward pointing selection arrow points to the right pointing arrow in the menu. The screen will look like this:



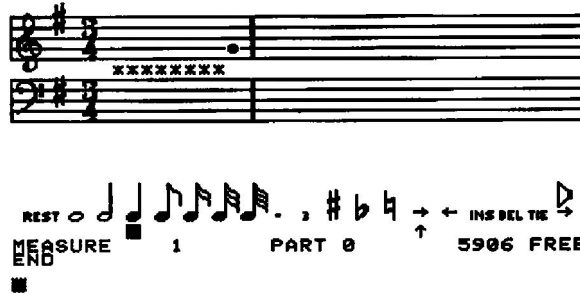
(The position of the flying saucer and the number of notes of FREE space available may be different than shown.) The right pointing arrow is used to move the cursor to the right. To cause a right movement, press the menu button using your left hand. The cursor will move right from an asterisk meaning QUARTER 240 to an asterisk meaning GAP 20. To move the cursor right again, press the button with your left hand again. The cursor now moves to TRANSPOSE 0. Press the button several times. The following items will appear: ATTACK 8192, DECAY 25, VOLUME 55000, SUSTAIN 0, and RELEASE 1500. Pressing the menu button again moves the cursor past the last of the 8 asterisks, and END appears under MEASURE 1 to indicate that the cursor is now at the end marker (that is, at the end of the song). This is where we will begin entering the notes of part 0. The screen should now look like this:



If it doesn't, you probably didn't start with RUN ENTRY like you should have. (The position of the flying saucer and the upward arrow, and the FREE number are not important.) Ready to really get into entering sheet music? Here's part 0 again, just as a reminder:



Using your right hand, turn the note paddle until the flying saucer is on the second line from the bottom of the treble staff, like this:



This is where the first note of part 0 should be. Still using your right hand, press the note button. A quarter note will appear at the second line, and the cursor will move over to the right. The pitch for that note is heard if you've got your synthesizer plugged in and your amplifier set up right. The screen now looks like this:



Normally when you type in something like TIME:3/4 or when you press the note button, the time signature (or note or whatever the cursor is pointing at) is written over and thus erased. However, erasing the end marker is not fun, so ENTRY automatically inserts the note (or whatever is entered) in front of the end marker. Then, when the cursor moves to the right, END is still shown under the MEASURE number since the end marker is still there.

It's time to give your left hand something to do for a while. Just for fun, position the arrow under the left pointing arrow in the menu (using the menu paddle, of course). Press the menu button. This will cause the cursor to move left. Under MEASURE 1, NOTE GN3 240 is displayed. That's the note you entered, a G Natural in the 3rd octave (the octave number is an ALF creation and has nothing to do with the rest of the world). "Natural!" means it is neither sharp nor flat. The 240 indicates the number of time periods long the note should be during playback. (When you press the note button to enter a note, it is just played for as long as you hold down the button.) Remember the QUARTER 240 that said quarter notes should be 240 time periods long? Well, they obviously are. Move the menu arrow so it is under the move right arrow and press the menu button. You're back to the end marker now. Isn't this exciting?

On to the second note. You've probably still got the note paddle set so the

flying saucer is on the second treble line. (If not, move it until it is.) Press the note button. The next note is heard and appears on the screen. It is the same as the first note. Now, turn the note paddle until the saucer moves up one click to the space above the second line. Press the button to enter this note (are you doing all this note-paddle stuff with only your right hand?). Not only do you hear this note and see it on the screen, but also a bar appears between the note and the flying saucer. This is because TIME 3/4 means that there are 3 (3/) quarter notes (/4) in a measure. Since the measure is now full, ENTRY automatically shows a measure bar. You'll notice that there is a bar at this point in the sheet music, too. If ENTRY and the sheet music don't seem to agree on where to put the bars, then either the sheet music has a typo (that is, a wrong note) or you've skipped a note or made some other error. Just by watching the measure bars you can be confident that you haven't made any timing mistakes.

If you're looking ahead at the music for part 0, then you know that the next note isn't a quarter note. It's a dotted quarter note, which plays for as long as a quarter note plus an eighth note. (A dot always means to add the time of the next shorter note to the note length shown.) You may well be wondering why ENTRY always makes a quarter note whenever you press down the note paddle button. Well, it's because a block is lit up under the quarter note in the menu. When you press the note button, a note as long as selected in the menu (shown by one or more blocks) is entered. To change from a quarter note to a dotted quarter note, you position the menu arrow under the dot, which is just to the left of the "3", and press the menu button (left hand, remember?). A block instantly appears under the dot, and the block under the quarter note remains. The screen now looks like this:



Okay, fire away. Move the note paddle down two clicks to the space under the second treble line, and press the note button. You see how you switch between the left and right hand, usually rotating a knob and pressing a button with the same hand? Since you generally keep your hands on the two knobs, you can enter notes really fast. You don't even have to look at the screen when you are entering several notes of the same length, because you can just count the "clicks" the Apple's built-in speaker makes at each line or space on the staff. (On some other music systems, you have to type in codes like the GN3 you saw on

the screen a while back, and this requires that you memorize the octave numbers.)

To enter the next note, position the menu arrow to the eighth note and press the button (I'm not going to remind you to use your left hand, since you've probably got that all straight by now). The blocks under the quarter note and the "dot" go out, and one appears under the eighth note, like this:



Move the note paddle up a click to the second line, and press the button to enter the eighth note. The screen now looks like this:



Let's take a look back. Move the cursor left one. (You know how to do it, we just did it a while back to see the first note displayed as NOTE GN3 240.) The eighth note shows up as NOTE GN3 120. It's the same as the first note in this part except it's half as long (only 120 time periods). That dotted quarter note we're coming up to should be a quarter (240) plus an eighth (120) long. Back up again to see it. Yep, NOTE FS3 360. But wait, doesn't FS3 mean an F sharp in the 3rd ALF octave? We didn't enter a sharp note. The reason for this is that the key signature indicates that all F's should be sharp. So, ENTRY automatically enters F's as being sharp, without the user having to specify it. Of course.

Back up three more times to get to the first note. Now, position the menu pointer to the rightmost menu item, a little speaker with a right arrow under it. Press the menu button, and a small block appears under the speaker/arrow. Curious? Position the arrow for right movement, and press the menu button five times to go past all the notes (do it fairly slowly, and pause a little extra at the dotted quarter note). You'll hear the first 5 notes of "America". The speaker/arrow activates playback during right movement. The timing of the notes

is still dependent on how long you press a button down, but don't worry. It won't be during actual playback. You don't believe me, do you? All right, type PLAY and press return. ENTRY shows "SET SPEED (255) AND PRESS BUTTON". Crank the menu paddle up all the way (it may not actually get up to 255, but who cares?). ENTRY doesn't happen to mention which button you should press, but it is the menu button. Trust me. Punch it and ENTRY will play the song. A little slow, perhaps, but we'll know better next time.

Let's put in another note. I'll bet you're thrilled at the prospect. Just select a quarter note using the menu paddle, flash the note paddle up to the space above the second treble line, and punch the note button. Here's a screen image just to make sure we're together:

The image shows two staves of musical notation. The top staff is in treble clef with a key signature of one sharp (F#) and a 4/4 time signature. It contains a sequence of notes: a quarter rest, followed by quarter notes G4, A4, B4, C5, B4, A4, G4. The bottom staff is in bass clef and is empty. Below the notation is a screen display showing: REST, MEASURE 3, PART 0, and 5900 FREE. A cursor is positioned under the third measure. A small square icon is located below the screen display.

Click up one to the third line. We're already set for quarter notes, so press the note button. Twice. Now, click up and press again (you should take a look at the music for part 0 again so you'll know what you're doing). That completes another measure. The display now shows MEASURE 4. This means the cursor is pointing to an item which is in the 4th measure. In this case, it is the end marker which is indeed in the 4th measure.

Faster now. Set for dotted quarter. Down a click and punch. Switch to eighth. Down a click and punch. Now quarter. Down a click, punch, up a click, punch, down, punch, down, punch. Last measure. Set for dotted half. (In case you haven't noticed, you can't set for "dot" and then "half" because "half" turns off "dot". Set "half" first, then "dot".) Okay. Up a click, and punch. We're out of music (just the first 6 measures, remember?). Are you getting fast at it yet? You will. It's easy. Let's see the screen now:

The image shows two staves of musical notation. The top staff is in treble clef with a key signature of one sharp (F#) and a 4/4 time signature. It contains a sequence of notes: a dotted quarter note G4, an eighth note A4, an eighth note B4, a quarter note C5, a quarter note B4, a quarter note A4, a quarter note G4, and a dotted half note G4. The bottom staff is in bass clef and is empty. Below the notation is a screen display showing: MEASURE 7, PART 0, and 5890 FREE. A cursor is positioned under the seventh measure. A small square icon is located below the screen display.

Type `PLAY` and press return. Let's try a speed of about 200 now. Adjust the menu paddle to some number in the vicinity of 200. (Don't get too picky, it's not important to get exactly 200.) Punch the button, and the first 6 glorious measures issue forth.

Rapture! Ecstasy! Sublime delight! (Where's my thesaurus?) Ah, the joys of music. And yet, that's just one part. Let's get on to `THREE PARTS`. Quick!

Fortunately, it is quick. First, we have to tell `ENTRY` that we want to add a second part. Type `EDIT` and press return. `ENTRY` responds by showing:



REST  # b h → ← INS DEL TIE 

MEASURE 7 PART 0 5890 FREE

NUMBER OF PARTS?

Since we want 2 parts, type 2 and press return. `ENTRY` then asks for the "suggested speed". Just press return to leave this as it was before. It will then display each of the four title lines. Just press return each time. The screen now shows:



REST  # b h → ← INS DEL TIE 

MEASURE 1 PART 0 5878 FREE

KEY IS

#

This is the beginning of Part 0, the part you just entered. The part just created is Part 1. To see Part 1, type `PART:1` and press return. The screen shows:



REST  # b h → ← INS DEL TIE 

MEASURE 1 PART 1 5878 FREE

#

This is just like Part 0 looked originally, except there are fewer notes of

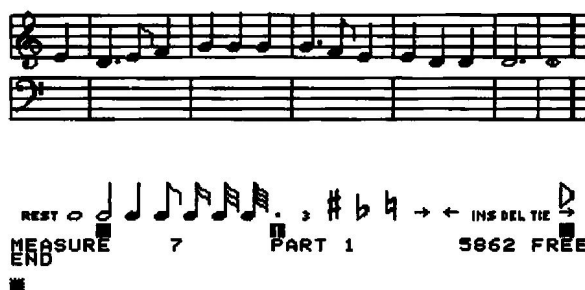
"free" memory, and the screen shows "PART 1" instead of "PART 0". You now proceed in the same fashion as before. Type KEY:1S (return) and TIME:3/4 (return). The music for Part 1 is as shown below:



Use the right arrow function to skip over the eight asterisks, and enter the first three notes as usual. The screen should now look like this:



Type PLAY and press return. (As usual, set the speed and press the paddle button to start playback.) You'll notice that only the first measure is played. Playback always stops when the end of the highest numbered part is reached. Since we've only entered the first measure in Part 1, and Part 1 is the highest numbered part, only the first measure is played. Enter the remaining notes of this part in the usual fashion. The screen will look like this:



Type PLAY and press return. (I won't tell you to adjust the playback speed paddle since you've got that figured out already.) If there are any wrong notes, back up and correct them. (More details on correcting wrong notes will be given later in this section.) You're now ready to enter the third part.

Type EDIT and press return. Ask for 3 parts this time, and then press return to skip the other questions. When Part 0 appears, type PART:2 to go to the third part. The screen shows:



Now enter the note. The sharp sign in the menu disappears into hyperspace:



Enter the rest of the part. The screen shows:



Type PLAY to hear the song and check for errors.

CORRECTING MISTAKES

Back up to the first note in measure 5 (of Part 2). Let's say we want to change this note so it is at the next space up on the staff. First, set the menu notes for a quarter note, and put the cursor in the space above the note:



The vertical position of the note paddle cursor is not important during a "tie" since the note paddle is not used. It is important to note that although the half note tied to a sixteenth note is shown as "two" notes, it is really only one. If you back up and look at it, you will see that the length shown is 540 time periods, which is a half (480) plus a sixteenth (60). In fact, the little curved line between notes always means that the multiple notes shown are really only one note. This happens on tied notes and on notes that have part of their duration in one measure and the remainder of their duration in the next measure. Tie in a sixty-fourth to the last note, and you'll see that more than two "notes" can be tied together to display a single note:



In general, mistakes are corrected (or any desired changes are made) by using the above functions (change a note, insert a note, delete a note, and tie additional duration to a note) until the screen shows what you want. When using these functions, only the current part is affected. In fact, the only functions available in ENTRY that affect anything besides the current part are the NEW, EDIT, STEREO, and SPEED commands which by their very nature must relate to the entire song.

ENTERING RESTS

On occasion a part must sit around for a while and not play anything. This is called a "rest". Rests are entered in much the same fashion as notes. There are two main differences: the vertical position of the note cursor doesn't matter (since rests don't have any "pitch"), and the menu paddle is used to enter a rest, rather than the note paddle. Obviously, you point the menu arrow to "REST" and press the menu button to enter a rest. The duration of the rest is determined by the menu, just as the duration of a note is. Rests are displayed with different symbols than notes. They correspond like this:



Let's start on a new song. (Actually, "song" refers to a musical composition with lyrics. Technically, one shouldn't use "song" to refer to just any melody, but

there isn't any simple word available. Musicians use "piece" or "work", apparently in an effort to avoid any disclosure that music is involved. In fact, all artists use "piece" and "work" to describe their creations.) Type NEW and press return. Press return 6 more times to avoid answering the useless questions. Skip over the key and time signatures, and the eight asterisks. Select quarter note, and press a REST. A quarter rest appears on the screen.



Now select sixteenth note duration and tie it onto the quarter rest. Oddly enough, the screen shows:



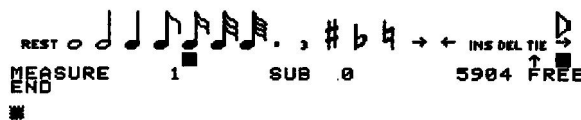
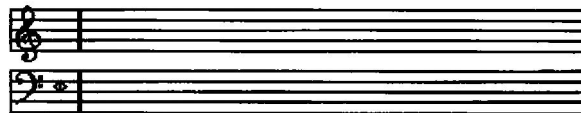
In traditional music notation, rests are never shown as being tied. This is because there is no difference between, for example, a half rest and two quarter rests during performance. The ENTRY screen display makes no distinction between a rest which is as long as a quarter plus a sixteenth, and two rests the first of which is a quarter and the second of which is a sixteenth. However, it takes only one "right movement" to skip a single tied rest, and two to skip past two individual rests. (Plus, two rests would take twice as much memory as a single rest.) Incidentally, when a large number of rests are tied together (for example, in a part which doesn't begin playing until far into the song) the cursor will be at the last of the rests displayed, and the measure number will reflect the measure number the rest starts in. (This is true of notes, too.)

SUBROUTINES

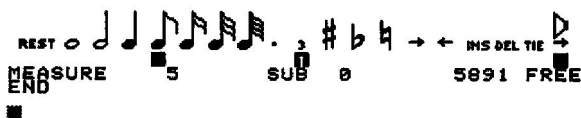
Most people are familiar with the song "Row, Row, Row your Boat". If you're not, become so. This song plays the same theme several times, and from several parts. It seems that one would have to enter this theme several times. Since repeated sections such as this are common in music, ENTRY has special provisions for entering them. The sheet music for this song is thus:



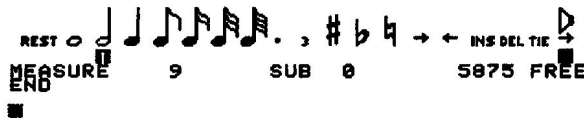
This theme must be entered in a special fashion which allows it to be played many times. This is done using a subroutine. Type NEW and press return several times (as usual) to start fresh. Now type SUBROUTINE:Ø and press return. The screen will show:



Type KEY:2S and TIME:2/4 to enter the key and time signatures. (Otherwise KEY:C and TIME:4/4 are assumed.) Enter the first four measures of the theme in the usual fashion. You'll notice that the next note is a triplet. Triplets are entered in the same fashion as dotted notes. Just light up the block under the "3" after selecting eighth note. Now press the note paddle button to enter the note. The screen will show:



The little 3 above the note indicates that it is a triplet. Conventional sheet music notation shows triplets with a curved arc above the three notes and a single 3. ENTRY puts a little 3 above each note. This is because ENTRY, unlike conventional notation, allows the presence of a single triplet note (that is, a single note with a duration equal to one of the notes of a conventional triplet set). Press the note button twice more to enter the remaining two triplet notes of that pitch, then enter the remaining three sets of triplets, and the rest of the theme. The screen will show:



Now type `PART:Ø` and press return to go to Part Ø. Type `KEY:2S` and `TIME:2/4` as usual, and skip the 8 asterisks. Now type `CALL:Ø` and press return. A 9th asterisk appears. During playback, this `CALL` causes the theme entered into its associated subroutine to be played. (`CALL:1` would play the theme entered into `SUBROUTINE:1`.) Type `PLAY` and press return. The basic theme is played. Now, type in another `CALL:Ø` after the first one. Type `PLAY` again and note that the basic theme is played twice.

Now `EDIT` the song to 2 parts. Type `PART:1`, `KEY:2S`, and `TIME:2/4`. This time, instead of skipping the 8 asterisks, step forward until `TRANSCOPE Ø` is shown. If we played the basic theme exactly the same in both parts, they would be hard to tell apart. So, type `TRANSCOPE:24` and press return. The `TRANSCOPE Ø` is of course thus changed to `TRANSCOPE 24`. The transpose function raises all following pitches by the specified amount of quarter steps. There are 24 quarter steps per octave (2 quarter steps is the difference between two adjacent keys on a piano, including both black and white keys), so `TRANSCOPE:24` will cause this part to be played one octave higher in pitch than the other part. Skip over the remaining asterisks. Part 1 is supposed to begin after Part Ø has already been playing for two measures. Select a whole note duration and enter a rest. It will show as two half rests due to the 2/4 time signature. Now type in two `CALL:Ø`'s. Type `PLAY`. A two-part round will be played.

Let's add a third part. `EDIT` the song to 3 parts. Type `PART:2`, `KEY:2S`, and `TIME:2/4`. Skip to the `TRANSCOPE` setting again. Let's shift this part down one octave. Oddly enough, to transpose down you take the number of quarter steps you wish to transpose down, and subtract that number from 256. 256-24 is 232, so type `TRANSCOPE:232`. Now skip past the other asterisks. Punch in a whole rest, then press `TIE` twice to make it two whole rests (which will display as four half rests, again due to the time signature). Type in the usual two `CALL:Ø`'s. Now just type `PLAY` to hear the full three-part round.

Perhaps you've noticed that you really didn't need the `KEY:2S`'s in the three parts, since there aren't any notes anyway. You could have simply deleted the key signature if you prefer. However, often there are notes in the part, and in that

case the key signature would be needed. In this particular instance, even the time signature could have been deleted without affecting the song. Naturally, the KEY:2S was needed within the subroutine, else the notes of the song would be incorrect.

Here are a few things you should know about subroutines. You can have 100 subroutines numbered 0 through 99. Always begin with subroutine 0 and proceed by 1's. If you press RESET, or if you save a song and load it again, all the subroutine numbers will be readjusted so they do begin with 0 and proceed by 1's. A subroutine is created when the first SUBROUTINE command using its number is entered. All subsequent SUBROUTINE commands with that number merely cause the subroutine to be displayed and to be available for editing. (That is, the first SUBROUTINE command for any given subroutine is like the EDIT command for new parts. All future SUBROUTINE commands are like the PART command for parts.) Once created, a subroutine cannot be destroyed. The most you can do is delete everything in it. A CALL can be entered only to an existing subroutine. (That is, you can't even enter a CALL to a subroutine you haven't created yet.) Subroutines are not limited to notes and rests. You can put a TRANSPOSE function in a subroutine, for example. Some things, like key and time signatures, can be put in a subroutine to affect the notes entered in the subroutine, but they do not affect the notes entered outside the subroutine, even after a CALL to the subroutine. The summary of commands in this section tells the effects of each command.

Subroutines can be used in a much more complex fashion than shown in this simple example. For example, subroutines can contain CALLs to other subroutines. If a subroutine contains a CALL to itself, the song will repeat forever (unless the highest numbered part does not use a subroutine which CALLs itself, in which case the song will stop whenever the highest numbered part stops). NOTE: be sure there is at least one note or rest in a subroutine that CALLs itself; otherwise the playback routines will not continue processing all parts.

LOADING AND SAVING SONGS

If you want to save Row, Row, Row then you should type SAVE and press return, if you want to save it on cassette tape. When saving a song to disk, it is necessary to specify a name. For example, you could type SAVE:ROW and press return. Names can contain any characters except comma, and can be up to 28 characters long. (Control letters and trailing spaces are ignored.) Disk specifications like ",D2" or ",S3,D2" can be added after the name if needed. Note that songs will appear in the catalog as Integer BASIC programs (even if your system doesn't have Integer BASIC) and will have names that begin with "M:". Songs are loaded the same way, using LOAD instead of SAVE.

The synthesizer is supplied with a few sample songs which can be loaded and played. Additional songs are available at extra cost.

ADJUSTING THE TEMPO

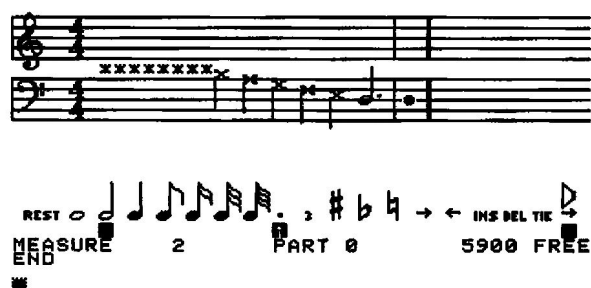
Let's say we want to enter the "row" theme to play twice as fast with the same paddle setting. That means each note will have to play for half as many time periods. Type NEW and press return as required, enter the key and time signatures, and you'll be at the QUARTER 240 function. Type QUARTER:120. This will make all quarter notes be entered as 120 time periods instead of 240 (and thus take half the time, so the song will play twice as fast). The other menu notes' duration values will change proportionately. Skip over the other asterisks and enter the theme. Now type PLAY and use the same paddle setting as you did previously. The song does indeed play twice as fast. Type PART:0 to get back to the beginning of the part, and skip over to the QUARTER function. Change it back to QUARTER:240. You'll notice that all previously entered notes show as notes half as long as originally entered. Examine any note by moving the cursor to it. Notice that the length in time periods is still the same. You didn't change any of the notes, only the QUARTER function, so of course none of the notes have been altered. Obviously ENTRY stores notes based on their "time period" length, and just computes the proper note to display based on the QUARTER setting. (And the QUARTER setting determines the "time period" length of notes when they are entered.) Since none of the notes have been changed, the song will still play as it did before. In fact, you can skip right a measure or two (you might want to look up the MEASURE command in the summary of commands) and insert a QUARTER:120. Notes before the QUARTER function will be shown as half as long as originally entered due to the QUARTER:240, and notes after the QUARTER:120 will be shown as entered. None of this affects playback, but any new notes you might enter would be based on the current QUARTER setting. Remove the inserted QUARTER, if you put one there, and change the QUARTER at the beginning to QUARTER:120 as it was when the notes were originally entered. Now type SPEED:2 and press return. This will multiply the "time period" lengths of all notes in all parts and subroutines by 2. Rest durations and QUARTER settings are also multiplied by the specified amount. Now the song plays twice as slow (also known as half as fast). In fact, it should look just like the original QUARTER:240 version, except that it used a subroutine and multiple parts. (CAUTION: the SPEED command can be tricky to use. See the complete description in the summary of commands.)

By typing in a QUARTER function wherever you need a different tempo, you can make the song play at different speeds from section to section. Just remember that the QUARTER function affects only notes which haven't been entered yet. Another way to get unusual note durations is by using the LENGTH command. Let's

say you want to play five notes in the space of a single quarter note. A standard quarter note is 240 time periods long, so each of your five notes will have to be 240/5 or 48. Unfortunately, there aren't any menu notes that are 48 time periods long. So, type LENGTH:48. The block(s) under the menu notes disappear to indicate a non-standard note length. All notes (and rests) you enter now will be 48 time periods long. Give it a try by making a new song and punching in five notes. The screen should look like this:



Since there is no representation for a note 48 time periods long, each note has a small X. To cease entering non-standard notes or rests, just activate any menu note. For example, put the menu arrow under the half note and press the menu button, then do the same for the dot (".") to select a dotted half note. Punch in a note, and the screen shows:



The measure bar shows that a full 4 quarter notes worth of duration have occurred, verifying that the five funny notes took up one quarter note of time.

ENVELOPES

Envelopes are a little complicated, and to really get the most out of your synthesizer is going to require a little study, some effort, a fair amount of calculation, and an awful lot of experimenting. Let's start at a very simple level. "Envelopes" are volume contours of each note. Since the word "volume" is used to mean the volume over several notes, we use the word "loudness" to refer to changes within a note. Both "volume" and "loudness" refer to the strength of the sound signal, but volume is used for long time durations (over several notes), and loudness is used for shorter durations (usually during a single note). "Amplitude" is used for the strength of the signal at any given instant, but this does not concern envelopes.

Let's begin with a typical note, one which begins with a loudness of zero (no sound) and also ends with a loudness of zero. In a simple sound, say that of a plucked string, the loudness rises very fast when the string is first plucked. Then, as the string vibrates, the loudness slowly dies out. The rising part of the envelope is traditionally called the "attack" stage. The falling part (where the loudness dies out) is called the "decay" stage. The whole envelope is called an AD (attack-decay) envelope.

In an AD envelope, there are three parameters. The first is the "attack rate", which is how fast the loudness goes from zero up to its highest point. The second we call the "volume level", which is the loudness level at the highest point. The third parameter is the "decay rate", which is how fast the loudness goes from the volume level back down to zero. For an AD envelope, the attack rate is usually very high (very fast). Plucked strings, for example, reach maximum loudness almost instantly. The decay rate can be varied for different sounds. A relatively fast decay creates a quick pluck for an instrument which decays quickly, like a banjo for example. A relatively slow decay creates a sound which dies out very slowly, more like a piano while the key is held down.

A more complex envelope is the ADSR (attack-decay-sustain-release) envelope. In the ADSR envelope, the attack stage is the same as in the AD envelope just described. However, the decay stage does not necessarily drop down to zero. Instead, it drops down to a selected level, called the "sustain level". Usually the sustain level is very high, nearly as high as the volume level. The loudness remains at the sustain level until something causes the "release" stage to begin. Usually the release stage begins a certain time before the next note. The release stage is the same as the decay stage of an AD envelope, except it drops from the sustain level (rather than the volume level) to zero. You'll notice that an AD envelope is just an ADSR envelope with a sustain level of zero. ADSR envelopes are useful for instruments which can play a note at a high volume level throughout the note, such as woodwind and brass instruments, or organs. The attack-decay stage of the ADSR envelope is used to give the sound an initial "thump" when desired. If the thump is not desired, the sustain level and volume levels are set the same.

Envelopes are controlled by the ATTACK, DECAY, SUSTAIN, RELEASE, VOLUME, and GAP commands. Both VOLUME and SUSTAIN specify a loudness level. SUSTAIN:Ø selects a very low level (soft), and SUSTAIN:65535 selects a very high level (loud). ATTACK, DECAY, and RELEASE specify a rate of change. ATTACK:Ø selects a very slow increase rate, and ATTACK:65535 selects a very fast increase rate. (Actually, 1 is very slow. Ø is stopped, or no change.) A blank song created with the NEW command contains some envelope settings which are useful for testing songs. Usually you enter the basic notes of a song, play around with the tempo

(playback speed) if necessary using SPEED commands and/or different QUARTER settings, and once you're satisfied with the tempo you go on to the envelope settings. This is because the SPEED command doesn't change any of the envelope settings. If you perfected your envelope settings and then used a SPEED command, the envelopes would no longer be perfect. This is needlessly complex to correct, so it is best to get the tempo going right before starting in on envelopes.

To change the initial envelope settings, just position the cursor at the appropriate item and type in a new value. For example, if you wish to have a slower attack rate, you might position the cursor at the ATTACK 8192 and type ATTACK:7800. Few songs use the same envelopes on all parts or even the same envelope throughout any particular part. At any point in a part, you can just "insert" new envelope parameters. During playback, the most recent setting (for each part) is used for envelope production. Since there are notes (and rests) between one envelope specification and another, the playback routines will not "see" the later specifications in the part until the note before them is finished. When they finish a note, they look at the next thing in the part. If it's not a note or a rest, they make whatever change is requested (a new attack value, for example) and then continue with the next thing in the part (until a note or rest is finally found).

Usually, on a synthesizer or a piano, the sustain stage ends (and the release stage begins) whenever the key being pressed is released (hence the word "release", obviously). There aren't any keys to release in the music data. So, the GAP function is used. It is used to specify how long before the end of the note the release stage should begin. For example, using QUARTER:240 settings, a whole note (960 time periods) played with a GAP setting of 240 would have three quarter notes (960-240, or 720 time periods) worth of attack, decay, and sustain; then one quarter note (240 time periods) worth of release. A rest automatically starts the release stage if it wasn't already. Notes shorter than the GAP setting have no release stage unless followed by a rest. GAP:65535 is used when no automatic release stage is desired.

Now is the time for all good men to experiment with envelope settings. Don't come back to this manual without experimenting for at least 7 million time periods.

You are now ready for the serious explanation of envelope production. Although theories change from time to time, today's leading scientists in enveology agree on the "wandering loudness" explanation. This one seems to fit the reality of the synthesizer most closely. The two main ingredients of this are "current loudness" and "desired loudness". The current loudness refers to a number which

ranges from 0 to 65535. This number divided by 4096 is the actual volume setting on the synthesizer at the moment. The desired loudness is also a number from 0 to 65535. The current loudness is "attracted" to the desired loudness, so it attempts to get closer and closer to it. Once each time period, the current loudness can increase by an amount less than or equal to the attack setting, or it can decrease by an amount less than or equal to the "current decay" setting. (Not to be confused with the "decay setting".) In this fashion, it will arrive at the desired loudness as quickly as the attack/current decay settings permit. Once the current loudness collides with the desired loudness, the desired loudness spontaneously changes to a new value, called the "current sustain level" (not to be confused with the "sustain setting"). Probability states that the new desired loudness may be different than the current loudness (although the current loudness is equal to the old desired loudness), so the current loudness must again seek the desired loudness. This astounding natural process continues at all times during playback. The current loudness cannot be affected directly, so it must be "guided" by selecting appropriate parameter settings.

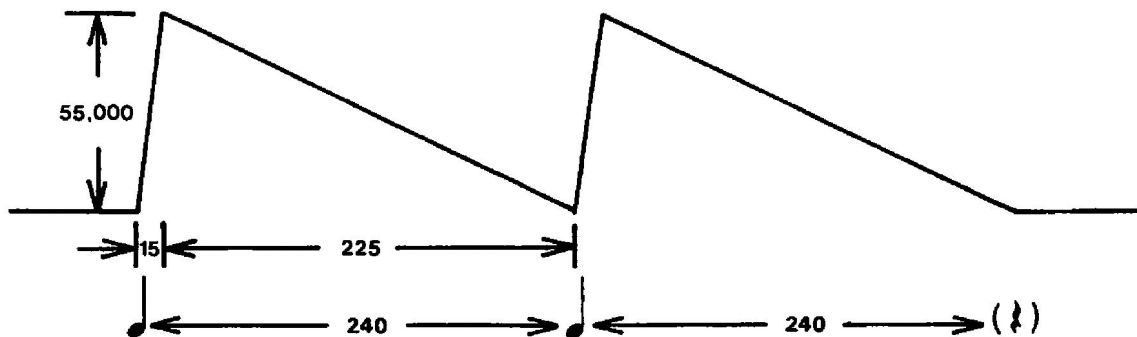
Notetrinos generated using a high-power paramatron at the University of Northern South Dakota (just across the border from Hoople) have revealed the following characteristics of these settings. (What?) When a new note begins, the most recent decay setting is written into the "current decay" rate, the most recent volume setting is written into the "desired loudness", and the most recent sustain setting is written into the "current sustain". This causes the attack and decay stages of the envelope to occur, since the current loudness (and thus the synthesizer volume) will raise (at the attack rate) to the selected volume level, at which time the sustain level becomes the new desired loudness, causing the current loudness to drop to the sustain level (at the decay rate). Once the sustain level is reached, the desired loudness stays constant (since it is equal to the current sustain setting which would normally become the new desired loudness) and thus the sustain stage of the envelope occurs until something changes.

Something changes when either (a) the time remaining for the current note equals the most recent GAP setting, (b) a rest is encountered, or (c) a new note is encountered. Case (c) has already been discussed (above). In either case (a) or (b), the release stage must begin. This is done by writing the most recent release setting into the "current decay" and a zero into the "desired loudness" and "current sustain". The current loudness (and, again, thus the actual synthesizer volume) then naturally drops to zero at the selected release rate.

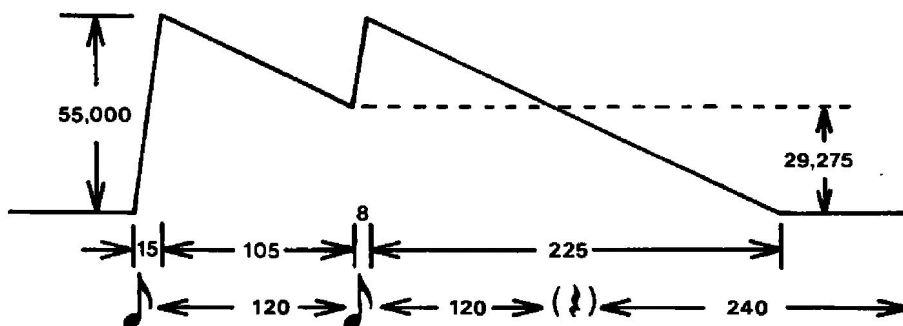
This simple process generates a variety of complex envelopes, for single notes or for several. Be ye not confused: each note does not necessarily have an "attack" and "decay" stage (and so forth). In fact, if the current loudness is

greater than the latest volume level when a new note begins (for example, the volume setting was just lowered drastically before this note, and the previous note had been at a very high volume with too slow a decay/release rate to drop very far), the note would begin with a "decay" stage, since the current loudness would have to go down to intercept the desired loudness (which would be the new volume level). Thus, the envelope parameters are not limited to a single note. In general, however, one will arrange the parameters so the envelope will be limited to a single note.

Some examples are in order. Let's say we want a simple AD (attack-decay, or "ping") envelope with a volume level of 55000. Further, let's say it is a quarter note with standard QUARTER settings (240 time periods) and we want the first 16th of the note to be the attack stage, and the remaining 15/16ths to be a full decay. The attack rate will have to be designed to take the current loudness from 0 to 55000 in 240/16 time periods. $55000 / (240/16)$ is 3666.67 so we want an attack setting of 3667. The decay rate will have to take the current loudness from this peak of 55000 back down to 0 in $240 * 15/16$ time periods. $55000 / (240 * 15/16)$ is 244.44 so we want a decay setting of 245. The loudness contour will appear thus:

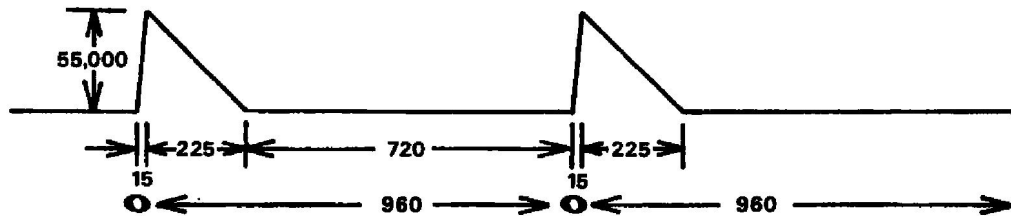


The GAP setting must be 65535 to avoid a release stage. Now, what if we played an eighth note with this setting? The loudness contour would appear thus:

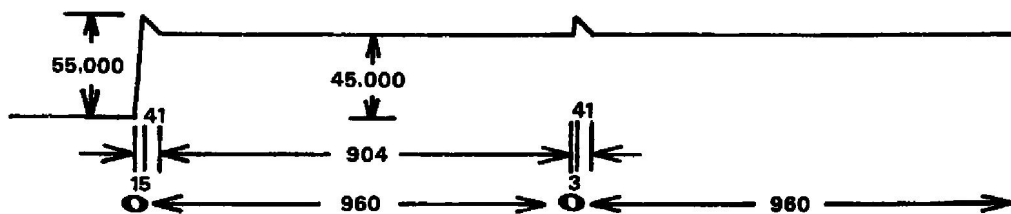


If an eighth note is followed by a rest, the release stage will begin. Therefore

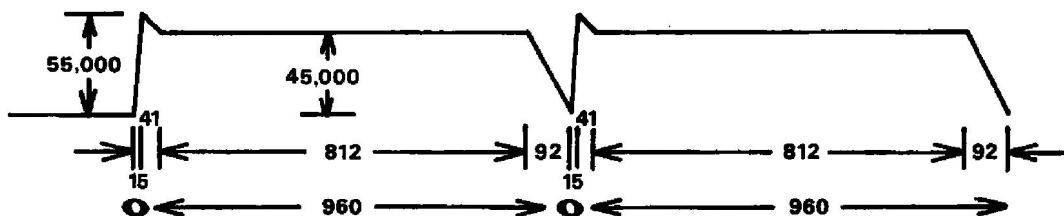
the release setting should be set to the same as the decay setting, unless you want something different to happen on notes followed by rests. What if we played a whole note? Behold:



This assumes the sustain level was set to 0. What if it were 45000?:



This is almost an ADSR (attack-decay-sustain-release) envelope. All we need is release. Let's say we want it to take half as long to release as the quarter note example took to decay. That means we'll need a release rate which is twice as fast, or 2×245 which is `RELEASE:490`. Now, it will take $45000/490$ time periods for the current loudness to drop from 45000 (the sustain level) to 0, so we need a `GAP` setting of $45000/490$ (which is 92) or greater if we want the release to go clear down to zero. That looks like this:



The sustain level need not be less than the volume level. For example, with a sustain level equal to the volume level, you get an attack-sustain-release envelope (organ like, using fast attack and release rates).

Experiment more with the settings. Draw graphs like the ones above if they help you. Look at other people's envelope settings if you run out of ideas. Here's a real tip: program what would normally be a whole part into a subroutine instead. Then you can call it from two parts, and use different envelope settings on each part (don't put envelope settings in the subroutine!). This will let you make more complex sounds, especially using different transpose settings or by putting a

short rest before the CALL in one of the parts to delay it slightly (for an "echo" effect) or both.

RECOMMENDED READING

For those of you who are unfamiliar with standard sheet music notation, or for those who encounter some particularly obscure notation, there is an excellent book which you can order from any bookstore. Just ask your local store to order "Music Notation, A Manual of Modern Practice" by Gardner Read, Taplinger Publishing Co. ISBN 0-8008-5453-5. In the unlikely event that you have no local bookstores, you can order it from ALF (part number 11-2-1).

BEAMING

IF WRITTEN:

ENTER:



etc.

J. S. Bach

Bist du bei mir, geh ich mit Freu-den

The image shows a musical score for J.S. Bach's 'Bist du bei mir'. It consists of three staves: a vocal line in treble clef and two piano accompaniment staves (treble and bass clefs). The key signature is B-flat major (two flats) and the time signature is 3/4. The lyrics are written below the vocal line.

BECOMES:

PART: 0

*
(additional *'s omitted for clarity)

The image shows a single musical staff labeled 'PART: 0' in treble clef, 3/4 time, B-flat major. It contains the first four notes of the vocal line from the original score. An asterisk is placed below the first note, and a note below it explains that additional asterisks are omitted for clarity.

PART: 1

*

The image shows a single musical staff labeled 'PART: 1' in treble clef, 3/4 time, B-flat major. It contains the next four notes of the vocal line. An asterisk is placed below the first note.

PART: 2

*

The image shows a single musical staff labeled 'PART: 2' in treble clef, 3/4 time, B-flat major. It contains the next four notes of the vocal line. An asterisk is placed below the first note. The staff is followed by a dashed line, indicating that the notes are not shown in this part.

PART: 3

*

The image shows a single musical staff labeled 'PART: 3' in treble clef, 3/4 time, B-flat major. It contains the next four notes of the vocal line. An asterisk is placed below the first note.

PART: 4

*

The image shows a single musical staff labeled 'PART: 4' in bass clef, 3/4 time, B-flat major. It contains the next four notes of the bass line from the original score. An asterisk is placed above the first note.

SAMPLE SONG BREAKDOWN

SUMMARY OF COMMANDS

ENTRY has four types of commands. They are:

1. Commands which are done immediately and have no effect on the song data.
2. Commands which are done immediately and have an effect on the song data.
3. Commands which are stored in the song data and do not affect playback directly.
4. Commands which are stored in the song data and do affect playback directly.

All commands, except those entered using the paddles, are typed in using the Apple keyboard in the following fashion. Each command has a "keyword", for example NEW or VOLUME. Some commands have one or more parameters, in which case the keyword is followed by a colon (:) and the parameter, for example VOLUME:55000. Thus, a command is always entered by typing the keyword and pressing return; or by typing the keyword, a colon, one or more parameters, and pressing return. (Do not type any spaces.) Since the keyword is always followed by a return or a colon, ENTRY has been written to allow abbreviation of the keyword. You can shorten any keyword as much as you like, as long as there are still enough letters to tell it apart from any other keyword. For example, INTEGER can be shortened to just I since no other keyword starts with I. SUBROUTINE can be shortened to SUB, but not to SU since it could then be either SUBROUTINE or SUSTAIN. An example of a complete abbreviated command is SUB:0 instead of SUBROUTINE:0. The right and left arrows on the Apple keyboard can be used to backspace and to forward space for error correction. When return is pressed, only letters to the left of the flashing cursor are considered part of the command, other letters are ignored. Control X can be used to clear the line and start over.

In the bold type for each command, anything inside <broken brackets> is an explanation rather than something to be typed literally. Anything inside [brackets] is optional.

TYPE 1 COMMANDS

These commands are done immediately. The song data is not changed at all.



The seven note duration symbols, plus "." and "3", are used to select a new note entry duration. (See REST and PADDLE 1 under Type 4 Commands.) They are requested by pressing Paddle 0's button while the upward-pointing arrow is aiming at the desired symbol. When one of the seven note duration symbols is requested, a block is lit under it. All other blocks under note duration symbols

(including "." and "3") are turned off. When "." is requested, the block under it changes (becomes lit if it wasn't, or is cleared if it was lit). When "3" is requested, the block under it changes.



The three accidental control symbols are used to select accidental control for future note entry (see PADDLE 1 under Type 4 Commands). They are requested by pressing Paddle 0's button while the upward-pointing arrow is aimed at the desired symbol. When one of the accidental control symbols is requested, the block under it is changed (becomes lit if it wasn't, or is cleared if it was lit) and the blocks under the other two accidental control symbols are cleared.



The left and right movement controls are used to move the cursor left or right. They are requested by pressing Paddle 0's button while the upward-pointing arrow is aimed at the desired symbol. When one of the movement control symbols is requested, the cursor will move one item in the indicated direction. Movement to the left of the first item in a subroutine or part is not allowed. Movement to the right of the end marker in a subroutine or part is not allowed. When a movement is requested which is not allowed, the request is ignored and the Apple speaker will beep.

INS

The insert symbol is used to turn insert mode on or off. It is requested by pressing Paddle 0's button while the upward-pointing arrow is aimed at INS. When requested, the block under INS is changed (becomes lit if it wasn't, or is cleared if it was lit). "Insert mode" is on when the block under INS is lit, or when the cursor is at the end marker of a part or subroutine. All Type 3 and Type 4 Commands are affected by insert mode.



The speaker/arrow symbol is used to select playback during forward (right) movement. It is requested by pressing Paddle 0's button while the upward-pointing arrow is aimed at the speaker/arrow symbol. When requested, the block under the symbol is changed (becomes lit if it wasn't, or is cleared if it was lit). When lit, notes moved past with the right movement symbol, and notes deleted with the DEL symbol, are sounded through the synthesizer.

GOTO:<0-8>

The GOTO command is equivalent to the PART command (a Type 1 Command) except

that a MEASURE command (a Type 1 Command) is automatically performed after the indicated part has been selected. The measure number used for the MEASURE command is whatever measure number was displayed on the screen at the time the GOTO command was entered. Sample command: GOTO:1 (return).

INTEGER

The INTEGER command is used to exit ENTRY and return to BASIC. The current song data is lost. ENTRY cannot be run again without first being reloaded. Note that when using the APPLESOFT version, the INTEGER command is used to return to BASIC, but APPLESOFT BASIC will be returned to rather than Integer BASIC.

LENGTH:<Ø-65535>

The LENGTH command is used to select a non-standard note duration. (See PADDLE Ø and PADDLE 1 under Type 4 Commands.) When entered, all blocks under the seven note duration symbols and under "." and "3" are cleared. The indicated duration is saved for future note and rest entry use. Sample command: LENGTH:48 (return).

MEASURE:<Ø-65535>

The MEASURE command is used to view a particular measure within a part or subroutine. The cursor moves to the first item within the specified measure number. MEASURE:Ø is equivalent to MEASURE:1. If no such measure exists, the cursor is moved to the end marker of the part or subroutine. Sample command: MEASURE:249 (return).

PART:<Ø-8>

The PART command is used to view a particular part (and thus select that part for possible editing). The cursor moves to the first item in the selected part, or to the end marker for that part if there are no items in the part. Sample command: PART:1 (return).

PLAY[:F]

The PLAY command is used to perform the current song (using a modified version of the PERFORM program). A simple low-res color display is shown during playback. In this display, each part has a blue horizontal line. In this line is a yellow dot which marks the position of middle C for that part (this dot will not be present when playing very high pitched notes). This middle C marker slides left and right one or more octaves if necessary to show whatever pitch range is currently being used. Above the horizontal line, a block is shown which indicates the pitch being produced. Higher pitches are to the right of the display. The color of this block indicates the "current loudness" of the pitch as follows: Ø-4Ø95 black, 4Ø96-8191 magenta, 8192-12287 dark blue, 12288-16383 purple, 16384-2Ø479 dark green, 2Ø48Ø-24575 grey, 24576-28671 medium blue,

28672-32767 light blue, 32768-36863 brown, 36864-40959 orange, 40960-45055 grey, 45056-49151 pink, 49152-53247 green, 53248-57343 yellow, 57344-61439 aqua, 61440-65535 white (loudest). (Based on Apple's suggested color names; actual colors may vary.) Ignoring the fact that there are two colors named grey, each color represents one of the 16 different actual volume settings on the synthesizer. PLAY:F performs the current song using the PERFORM program (that is, with no display). NOTE: both PLAY commands change (a) the CHANNEL function settings and (b) the subroutine FE bytes. These changes will not be apparent to the ENTRY user, but could affect PERFORM users. See the PERFORM section for additional information. Sample command: PLAY (return).

SAVE[:<song name>[<disk specifications>]]

The SAVE command is used to write the current song data on cassette tape (or whatever might be connected to the Apple's cassette output jack) or on disk. SAVE saves the song to cassette tape. SAVE:<song name>[<disk specifications>] saves the song to disk. Both commands are used in the same fashion as the SAVE commands in BASIC. One exception: song names may contain 0 to 28 characters, including any character except comma (for any character, including the first); control characters and trailing spaces are ignored, but leading spaces are not. Sample command: SAVE:GALACTIC TRIUMPH,D2 (return).

*****DISK[:<comment>]**

The ***DISK command is not used but has been included for compatibility with the 10-5-16 Apple Music Synthesizer. For additional information, order the Apple Music Synthesizer owner's manual, order number 11-1-6.

TYPE 2 COMMANDS

These commands are done immediately. They do not cause an item to be written at the current cursor location, as Type 3 and Type 4 Commands do, but they do affect the current song data.

DEL

The DEL symbol is used to delete the item the cursor is currently at. It is requested by pressing Paddle 0's button while the upward-pointing arrow is aimed at DEL. When requested, the item the cursor is at is deleted from the song data. If it is a note, it is sounded through the synthesizer if the speaker/arrow block is lit (see the speaker/arrow Type 1 Command). The end marker of a part or subroutine cannot be deleted. If this is attempted, the Apple speaker beeps.

DELETE:<1-255>

The DELETE command is used to remove one or more items from the current part or subroutine. It is the same as one or more DEL symbol requests (above) except

the notes are never sounded and there is no "beep" when an attempt is made to delete the end marker. The number of DEL's is selected by the <1-255> parameter. More than 255 items can be deleted only using more than one DELETE command. Sample command: DELETE:73 (return).

EDIT

The EDIT command is used to increase the number of parts, change the suggested speed, and/or change any or all of the 4 title lines. Once entered, the command proceeds to ask for the new NUMBER OF PARTS?, SUGGESTED SPEED?, and TITLE LINE 1 through TITLE LINE 4. If there is no change desired on any item, just press return. Otherwise, enter the new value and press return. For each TITLE LINE, the current line is displayed and can then be edited using the left and right arrow keys on the Apple keyboard. Note that when return is pressed for a title line, all characters to the right of the flashing cursor, and the character under the flashing cursor unless it is the 40th character, are set to space. The SUGGESTED SPEED must be from 0 to 255. (1 through 255 select paddle speeds to be suggested before song playback. 0 causes the song to be played without waiting for paddle adjustment.) The NUMBER OF PARTS? must be greater than or equal to the current number of parts, but less than 10. If the number of parts is increased, the stereo settings are set to standard settings (see NEW, a Type 2 Command; and STEREO, a Type 2 Command). See SUBROUTINE (a Type 2 Command) for details on reduction of "notes free" when increasing the number of parts. The cursor is set to the first item in Part 0. Sample command: EDIT (return).

LOAD[:<song name>[<disk specifications>]]

The LOAD command is used to load a song from cassette tape (or whatever is connected to the Apple's cassette in jack) or disk. The song currently in memory is lost. These commands are used the same as the LOAD commands in BASIC. See SAVE (a Type 1 Command) for additional comments. The cursor is set to the first item in Part 0. Sample command: LOAD:GALACTIC TRIUMPH (return).

NEW

The NEW command is used to start fresh. Once entered, the NEW command asks for the NUMBER OF PARTS? which should generally be entered as 1. If return is pressed, 1 is assumed. The number of parts cannot exceed 9. Remember that parts created cannot be destroyed and that song playback ends when the end of the highest numbered part is reached. New parts (created either with NEW or with EDIT, a Type 2 Command) contain KEY:C, TIME:4/4, QUARTER:240, GAP:20, TRANSPOSE:0, ATTACK:8192, DECAY:25, VOLUME:55000, SUSTAIN:0, and RELEASE:1500. (All subroutines and parts always end with an end marker.) Stereo is set to the standard values: STEREO:MLRMLRM*L*R*. The NEW command then asks for the SUGGESTED SPEED? which can be given as any integer from 0 to 255, or just press return for 255. Finally, the NEW command asks for the 4 TITLE LINES.

These are initially set to all spaces. The cursor is set to the first item in Part 0. Sample command: NEW (return).

SPEED:<1-65535>[/<1-65535>]

The SPEED command is used to change the duration of all notes, rests, and QUARTER functions in all parts and subroutines. The colon after SPEED is followed by an integer from 1 to 65535 to multiply all time durations by. This is optionally followed by a slash (/) and another integer from 1 to 65535 indicating a number to divide by. (If not specified, this is assumed to be 1.) All time durations are multiplied by the first integer, then divided by the second integer. Any "remainder" (or non-integral portion) is ignored, and the result MOD 65536 is used. For example, a note length of 240 divided by 50 (using SPEED:1/50) would become 4 since $240/50$ equals 4.8. The .8 time periods dropped will eventually accumulate (differently in different parts) and create unusual timing. Therefore, such non-integral results should usually be avoided. Any 0 results are changed to 1. **CAUTION:** extreme care must be taken to avoid destruction of the song! Saving the song prior to attempting a SPEED command is strongly recommended. Also, see QUARTER, a Type 3 command. Sample command: SAVE:GALACTIC TRIUMPH (return) SPEED:1/2 (return).

STEREO:<string>

The STEREO command is used to change the stereo selection programmed in the song. The first letter in the string specifies the position for Part 0, the second for Part 1, etc. It must consist of L's (for Left), M's (for Middle), and R's (for Right). There cannot be more than 3 L's, 3 M's, or more than 3 R's.

Following the L, M, or R for each part, a star (asterisk, *) may be typed to allow fuzz (white noise). Any part using fuzz must be identified by a star following its stereo letter. For example, STEREO:ML*RM* sets parts 0 and 3 for "middle", part 1 for left, and part 2 for right; it also allows use of fuzz on parts 1 and 3. Only one L, M, and R can be followed by a star since fuzz can be used on only one channel per stereo position.

NOTE: the EDIT command changes the STEREO settings if the number of parts is increased. The stereo settings selected are programmed into the CHANNEL function (see the PERFORM section) and thus will be saved with the song. Sample command: STEREO:MLMR (return).

SUBROUTINE:<0-99>

The SUBROUTINE command is used to create a subroutine, or to view (and thus ready for editing) an existing subroutine. (Note: this command may be considered a Type 1 Command if used to access an existing subroutine rather than create a new one.) The creation of a new subroutine will reduce the number of free notes

by the following amounts depending on the number of parts: 2 for 1 part, 3 for 2, 4 for 3 or 4, 5 for 5, 6 for 6 or 7, 7 for 8, and 8 notes for 9 parts. (NOTE: increasing the number of parts with EDIT, a Type 2 Command, reduces the number of free notes by enough to account for the difference in storage requirements for each subroutine (since more "notes" of storage are required per subroutine when more parts are present, as shown above), plus 12 and 2/3rds notes per new part.) The cursor is positioned to the first item in the selected subroutine, or the end marker in that subroutine if there are no items. **CAUTION:** subroutines are assigned numbers from 0 up (by ones) when a song is loaded and when RESET is pressed (C00G must be typed on systems without an Auto-Start ROM). The numerical order of the subroutines does not change. Sample command: SUBROUTINE:83 (return).

TYPE 3 COMMANDS

These commands are not done immediately, but rather are stored in the song data at the current cursor position. The item currently at the cursor position is erased unless insert mode is on. These commands do not affect playback. They affect only newly entered notes and rests, or the screen display. Commands of this type included within a subroutine affect only the display and entry of notes within the subroutine itself, and not within any part (or other subroutine) calling the subroutine. The number of notes free goes down by 1 for each inserted command, but stays the same for replaced commands.

KEY:<1-6><S-F> or KEY:C

The KEY command is used to change the key signature. (If no KEY command has occurred in the part or subroutine so far, the key is assumed to be KEY:C.) KEY:C specifies no sharps or flats, and an integer from 1 to 6 followed by an S or an F specifies the indicated number of sharps (S) or flats (F). All notes entered so as to appear in the song data after this KEY command (but before the next KEY command) will be affected by this KEY command. Any note not entered as "sharp", "flat", or "natural" will be changed to sharp if it is one of the notes indicated as sharp in the key signature, or changed to flat if it is one of the notes indicated as flat in the key signature. Notes not indicated as either sharp or flat by the key signature are left as is. Sample command: KEY:3S (return).

QUARTER:<1-65535>

The QUARTER command is used to change the duration of notes entered except when using non-standard durations with LENGTH (a Type 1 Command). All notes entered so as to appear in the song data after this QUARTER command but before the next QUARTER command will be affected. (If no QUARTER command has occurred in the part or subroutine so far, it is assumed to be QUARTER:240. All subroutines should start with a QUARTER command if the SPEED command is to be

used.) See the PADDLE 0 and PADDLE 1 Type 4 Commands for additional details. Sample command: QUARTER:480 (return).

TIME:<1-19>/<note>

The TIME command is used to change the time signature. (If no TIME command has occurred in the part or subroutine so far, the meter is assumed to be 4/4.) The colon after TIME is followed by the number of notes (of a certain duration) to occur per measure. This is followed by a slash (/) which does not mean division (this is a special case). The slash is followed by an integer which specifies the note duration referenced by the other integer. It must be 1 for a whole note, 2 for a half, 4 for a quarter, 8 for an eighth, or 16 for a sixteenth note. The number of time periods allowed per measure will be the current QUARTER setting times 4 times the number before the slash, all divided by the number after the slash. This command determines the positioning of measure bars, which in turn affects whether a note is sharp (or flat) or not (see the PADDLE 1 Type 4 Command). It affects all notes entered so as to appear in the song data after this TIME command but before the next TIME command. Sample command: TIME:2/2 (return).

TYPE 4 COMMANDS

These commands are not done immediately, but rather are stored in the song data at the current cursor position. The item currently at the cursor position is erased unless insert mode is on. These commands are executed during playback. They are executed during a subroutine call and thus may effect notes entered in a given part (or subroutine) after a call to the subroutine containing these commands. The number of notes remaining goes down by 1 for each inserted command, and stays the same for replaced commands, except as noted for TIE. <value> always refers to an integer from 0 to 65535, optionally followed by a slash (/) and another integer from 0 to 65535. When the slash is specified, the indicated division is done and the resultant value (ignoring any remainder or non-integral portion) is used as the parameter.

REST

The REST symbol is requested by pressing Paddle 0's button while the upward-pointing arrow is pointing at REST. When requested, a rest is written in the song data. The duration of the rest is determined in the same fashion as the PADDLE 1 Type 4 Command (below).

PADDLE 1

Note entry is accomplished by pressing Paddle 1's button. The vertical position of the note cursor (controlled by Paddle 1's knob) determines the pitch of the note, subject to various sharps and flats, and (during playback only) the current

TRANPOSE (Type 4 Command) setting. Notes will be natural, sharp, or flat; as indicated by a block under one of these in the menu, and the blocks cleared, if one of these blocks is lit. Otherwise, notes are entered as natural unless they must be sharp or flat due to the current key signature or due to a prior note in the measure of the same pitch being sharp or flat. (Note: all octaves are affected by the key signature, but not by prior sharp or flat notes in the measure.) Natural, sharp, or flat signs are displayed on the screen only when necessary. Duration is as specified by LENGTH (a Type 1 Command) unless one or more blocks are lit under the seven notes in the menu. (Note: "." and "3" do affect LENGTH settings.) If a block is lit, the length will be assumed to be as specified by the most recent QUARTER command for quarter notes, and proportional values for all other notes. A block under "." multiplies the length by 3/2, and a block under "3" multiplies the length by 2/3. (A block under both multiplies the length by 2/3 and then by 3/2.) Entry of a sixty-fourth note (selected by a block under the sixty-fourth note) is not allowed if the "." block is lit. (Dotted sixty-fourth notes are never displayed.)

TIE

The TIE symbol is requested by pressing Paddle 0's button while the upward-pointing arrow is pointing at TIE. When requested, the duration which would be used if a note were entered (see the PADDLE 1 Type 4 Command) is added to the duration of the note or rest the cursor is currently at. (If the cursor is not at a note or rest, the Apple speaker beeps and the cursor moves left one item.) This command is unaffected by insert mode, and it never changes the number of notes free.

ATTACK:<value>

The ATTACK command changes the current attack setting. The value specified is the maximum amount the "current loudness" can increase in any given "time period". Sample command: ATTACK:55000/30 (return).

CALL:<0-99>

The CALL command is used to have the Type 4 Commands in the specified subroutine be executed during playback. The integer (from 0 to 99) specifies which subroutine should be done. More than one part may call the same subroutine (or different subroutines) at the same time. A subroutine may call itself provided at least one time period of duration occurs within the subroutine prior to the call to itself. A CALL cannot be entered until after its subroutine has been created. See SUBROUTINE (a Type 2 Command) for additional information. Sample command: CALL:83 (return).

DECAY:<value>

The DECAY command changes the current "decay setting". The value specified is the maximum amount the "current loudness" can decrease in any given "time

period" unless the RELEASE rate is currently being used. Sample command: DECAY:100 (return).

FUZZ:ON or FUZZ:OFF

The FUZZ command is used to select fuzz (white noise) mode or normal mode. FUZZ:ON selects fuzz mode, and FUZZ:OFF selects normal mode. (NOTE: Fuzz mode must not be used in a part unless it has been allowed by a star in the STEREO command. See STEREO, a Type 2 command.) In fuzz mode, pseudo white noise is produced rather than a simple square wave tone. When used with high pitches and fast envelopes, percussive bursts can be made. Note that the "normal" mode tone will also be produced when FUZZ mode is on, but at a constant volume which will be whatever the "current loudness" was when the FUZZ:ON command was found. To avoid a tone, be sure the envelope has fully decayed (to zero) before using FUZZ:ON. Similarly, to avoid "white noise" during normal mode, be sure the envelope has fully decayed (to zero) before using FUZZ:OFF. Sample command: FUZZ:ON (return).

GAP:<value>

The GAP command changes the current gap setting. When the time remaining for any note equals the current gap setting, the release stage of the envelope begins. Sample command: GAP:60 (return).

POKE:<0-255>,<0-255>,<0-255>

The POKE command is used to enter non-standard commands. **CAUTION:** use of this command renders this documentation meaningless and may well scramble memory during playback. Integers from 30 to 175 followed by 0 and 0 (for example, POKE:78,0,0) enter notes of zero duration; the correct duration can be TIEd in. For information on other values, see the PERFORM section, and the SONG DATA FORMAT heading in this section. Sample command: POKE:144,240,0 (return).

RELEASE:<value>

The RELEASE command changes the current release setting. The value specified is the maximum amount the "current loudness" can decrease in any given "time period" unless the DECAY rate is currently being used. Sample command: RELEASE:100 (return).

SUSTAIN:<value>

The SUSTAIN command changes the current "sustain setting". The value specified is the "desired loudness" which the "current loudness" follows, unless the desired loudness is currently 0 for a release stage or the current volume setting for an attack stage. Sample command: SUSTAIN:45000 (return).

TEMPO:<value>

The TEMPO command is not used but has been included for compatibility with the 10-5-16 Apple Music Synthesizer. For additional information, order the Apple Music Synthesizer owner's manual, order number 11-1-6.

TRANPOSE:<0-255>

The TRANPOSE command is used to change the current transpose setting. Values from 0 to 127 raise all following pitches (until the next TRANPOSE command) by 0 to 127 quarter steps; values from 255 to 128 lower all following pitches by 1 to 128 quarter steps. 24 quarter steps equals 1 octave. Sample command: TRANPOSE:232 (return).

VOLUME:<value>

The VOLUME command changes the current volume setting. The value specified is the "desired loudness" which the "current loudness" follows unless the envelope is not currently in an attack stage. Sample command: VOLUME:500000 (return).

TIPS

PARTIAL STARTING MEASURE

Often songs begin with a measure which is short, perhaps containing only a single note. If such a song were entered in the normal fashion, the measure bars would not appear at the correct places. There are many ways of solving this problem. The simplest and perhaps best way is to start by entering a rest which is long enough to fill one measure when the partial (starting) measure is entered after the rest. Not only does this put the measure bars in the right places, it also causes a brief delay before song playback begins during a PLAY command, which may be considered desirable. Another method is to put the partial measure in a subroutine, and call it. (The duration of notes within a subroutine is not added to a part which contains a CALL to that subroutine.) Yet another method is to enter the partial measure, and then enter a TIME or a QUARTER command to start the measure over.

RESTS AT THE END OF PARTS

Each part should end with a rest. It can be as short as you like, and it serves to begin the release stage of the envelope. Otherwise a release stage may begin unexpectedly (when the constantly cycling time remaining equals the current GAP size). Additionally, the highest numbered part should end with a rest long enough to let all parts decay (or release, actually) down to zero volume, and perhaps even show a "blank" screen for a second. PERFORM users may find this particularly necessary, lest the parts continue playing after PERFORM returns to the calling program.

PADDLE SETTINGS

Paddle settings which are too small will create "time periods" which are not long enough for all necessary calculations. When this happens, the "time period" is lengthened so that all calculations are completed. Since the calculation time required varies, the song playback speed will vary too. There is no time period variation when the paddle setting is high enough. Generally, paddle settings lower than 150 are never used. Songs having many parts active and using several levels of subroutines may require even higher settings. The number of time periods in one second is approximately $93000/(\text{<paddle setting>}+1)$.

"BACKUP"

While entering particularly long songs, it is a good idea to save the song periodically in case the power fails, ENTRY hits an undiscovered bug, or you accidentally delete half the melody.

TRANPOSE

Each part must contain a TRANPOSE before the first note, even if it is a TRANPOSE:0.

COPYING SONGS WITHOUT ENTRY

Systems equipped with Integer BASIC can copy songs from one tape or disk to another without running ENTRY. Just load the song as if it really were an Integer BASIC program, and save it. Since it isn't a BASIC program, attempting to change or delete a line, or attempting to RUN it, would probably scramble the song data; however, a load followed immediately by a save will work properly.

RESET

On systems without an Auto-Start ROM, C00G (return) must be typed if RESET is pressed. That's C zero zero G, not COOG. RESET can safely be used during a PLAY command. RESET must not be used during the execution of any other command, or the song data may be destroyed.

INTEGER/APPLESOFT SWITCH

On systems with a ROM card (for Applesoft or Integer BASIC), the switch must be set for a start-up language which matches the version of ENTRY being used.

SONG DATA FORMAT

Song data is stored as described in the PERFORM section with the following changes:

1. Song data always begins in memory at 5000 hex.
2. The END command (FF 00 00) is followed by a byte giving the suggested speed, then 160 bytes which form the four title lines.
3. The QUARTER command is stored with command type FB hex.
4. The KEY command is stored with command type FC hex. A parameter of zero indicates C. Otherwise, the number of sharps/flats is stored with the most significant bit being 0 for flat or 1 for sharp. The third byte is not used.
5. The TIME command is stored with command type FD hex. The second byte indicates the number of notes per measure, and the third byte the type of note.
6. All TRANPOSE commands have a third byte of FE. This allows the least significant bit of each note to indicate sharp or flat.
7. When loaded using Integer BASIC, locations CA and CB hex ("PP") indicate the starting address of the data. Locations 4C and 4D hex ("HIMEM") indicate the address past the last byte of data.

SELECTED HEX ADDRESSES

4C & 4D: defines the address of the first byte of unavailable memory

72: defines the slot number times 16

5E & 5F: defines the address of the first byte following the song's title lines (end of song pointer)

5000: start of song data

A9B: start of pitch divisor table

4F38: start of Entry-generated subroutine address table

4D6E-4D91: part initialization data

4EE8-4F36: command table expansion area

4DC0-4DCB: standard stereo positions

Base page usage: (see also PERFORM base page usage)

0-19 26-27 36-39 3C-3F 4A-4D 50-55 58-8E CA-CD

3

PLAY

The PLAY program is used to play songs entered with ENTRY. Songs can be read from cassette tape or from disk. Although songs cannot be edited with PLAY, it has several advantages over ENTRY. PLAY's main advantage is that it requires less memory than ENTRY. This means that PLAY can be loaded (from tape or disk) faster than ENTRY, and it allows playback of songs which are too large to load with ENTRY. Another important feature of PLAY is that most disk commands can be used (ENTRY allows only LOAD and SAVE). This allows "Exec Files" to be used, either as created by the DISCO program or custom files.

To run PLAY, you must have 5K bytes of memory plus enough additional memory to hold the song. If you are using a DISK II, you need 15.5K plus the song length. (Using the Applesoft version, these figures are 8K and 18.5K.) The maximum song length is 28K. (17.5K for songs entered using a DISK II system with MAXFILES 3.)

First, load the program from disk or cassette tape. List line 10. It will be 10 SLOT=4. Carefully retype the line changing only the digit 4 to the proper digit for your system. Now save the program on your disk. If you do not have a DISK II, save the program using your own recorder to improve loadability. The program is now configured for your system, and can be run any time you like without having to change line 10. If you ever change the slot position of your synthesizer, you should do this configuration procedure again.

When run, PLAY will print a period (.) as a prompt character. The following commands can then be used:

LOAD[:<song name>[<disk specifications>]]

This command is the same as the load command in ENTRY (see the ENTRY section, SUMMARY OF COMMANDS).

PLAY[:<song name>[<disk specifications>]]

This command is a mixture of the play command in ENTRY (see the ENTRY section, SUMMARY OF COMMANDS) and the load command (above). Typing PLAY (return) is used to play the song currently in memory (you must have already loaded a song, of course). PLAY:<song name>[<disk specifications>] is used to load a song and then play it.

STOP

This command is used only in ALBUM files created by DISCO (see the DISCO section). It goes to BASIC, leaving the PLAY program in memory for continuation with RUN. Either RUN or INT (FP when using Applesoft) should always be used after a STOP command.

INT or FP

INT (or FP for Applesoft) is used to stop using PLAY. The PLAY program is erased and must be reloaded if you desire to run it again.

Most disk commands, such as CATALOG and EXEC, can be used while running PLAY.

ENTRY's PLAY:F is not available in PLAY since the F would be assumed to be a song name.

If you wish to stop playback, press RESET. On systems not equipped with an Auto-Start ROM, type 3D0G (control C return on cassette systems) to return to BASIC. Once in BASIC, type RUN to clear the synthesizer and continue using PLAY.

4

DISCO

The DISCO program is used to create an "Exec File" which can be used to play songs in succession. It can also randomize the playback order. It can be used only on systems equipped with a DISK II. A text file named ALBUM is created, so a disk which is not write-protected is required. The procedure is as follows:

Load DISCO from cassette tape or disk, and save it on your disk. (If you have already done this, just LOAD the program from your disk.) (NOTE: there is no need to change line 10 for this program.) Type RUN 1000 and press return. DISCO will print a brief set of instructions.

It is best if you have a printed catalog listing for this next step. If you don't have one, just type CATALOG occasionally to see the catalog listing. Type in the song names to be played, pressing return after each song name. Do not type the "M:" (for example, if you used SAVE:GALACTIC TRIUMPH from ENTRY, then you should type GALACTIC TRIUMPH (return) for DISCO, rather than M:GALACTIC TRIUMPH which is how the song will appear in the catalog). If you wish to have the songs played in a particular order, you must type them into DISCO in that order.

When all songs have been entered, type STOP and press return. **CAUTION:** care must be taken to not hold down the keyboard keys while typing STOP. The lack of n-key rollover on the Apple keyboard will cause unseen control letters to be entered if several keys are held down at once. This would cause a song title to be entered which consists of STOP and these control letters, rather than a STOP command.

If you wish to always use the same playback order, type LOCK ALBUM and press return. It will be necessary to type UNLOCK ALBUM if you ever wish to delete the ALBUM file or make any changes to it.

To play the whole sequence (or "album") of songs, you type EXEC ALBUM and press return. If you wish to have the order randomized, type RUN DISCO (or, if DISCO is already loaded, type RUN). To do either of these, a properly configured PLAY program must be on the disk and named PLAY. When album playback is complete, you can type RUN to run PLAY (do not type RUN PLAY), otherwise type INT or FP. If you wish to hear the songs again, type EXEC ALBUM (or RUN DISCO) after typing INT or FP.

TO ADD SONGS

Load the DISCO program, and type RUN 2000 (return). After the instructions are printed, proceed in the same fashion as when originally creating the album (done with RUN 1000, above).

TO START OVER

If you wish to scratch the old ALBUM file and make a new one, type DELETE ALBUM (return). Then LOAD DISCO and RUN 1000 as described above. If you do not DELETE ALBUM, and if the new ALBUM file is shorter than the old one, commands remaining at the end of the file will result in errors after album playback is completed.

USING "START" and "END"

When randomizing the song order using RUN DISCO, you can have one particular song played as the first song, and/or another played as the last. These songs must be named START (for the first song) or END (for the last song). When a song named END is entered (during RUN 1000 or RUN 2000), DISCO stops (there is no need to use a STOP command). END will remain the last song even if more songs are added (using RUN 2000) or the order is randomized (using RUN). The song must appear in the catalog as M:END. The START song should generally be entered as the first song, when the album is first made using RUN 1000. Otherwise it will not be the first song until it is randomly placed as the first (but will remain first from then on). It must appear in the catalog as M:START.

USING MORE THAN ONE DISK DRIVE

Songs in an album can occupy more than one disk drive. The ALBUM file and the PLAY program must be on the same disk. Songs must be entered (when using RUN 1000 or RUN 2000) followed by the proper disk specification. For example, when using two drives on the same controller, all songs on drive 1 must be followed by ",D1" and all songs on drive 2 must be followed by ",D2". If you are not using the randomization feature, the disk specifications need only be given when there is a change (for example, when the previous song was on drive 1 but this song is on drive 2, it must be followed by ",D2"). Be sure to leave enough room on the disk containing the ALBUM file for possible expansion of the file.

5

PROGRAMMING

WITH PERFORM

The PERFORM program is used to play songs from your own programs. It can play songs entered with ENTRY, or songs created by other means (see the SONG DATA description in this section).

PERFORM is rather difficult to use on systems which do not have a DISK II. In this case, PERFORM must be loaded from tape and RUN. PERFORM will then be located at 2050 decimal (802 hex) in memory. LOMEM is automatically changed so PERFORM will not be erased by other programs you may load (note: be sure to avoid using control B or programs which change LOMEM). To use PERFORM, you must have a song in memory. At 2048 decimal (800 hex) you must put the starting address of the song MOD 256. (In Applesoft, this is $\text{address} - \text{INT}(\text{address}/256) * 256$ since MOD is not available.) At 2049 decimal (801 hex) you must put the starting address of the song divided by 256. (In Applesoft, this is $\text{INT}(\text{address}/256)$.) Then, a CALL to 2050 decimal (802 hex) causes the song to be played. The remainder of this section assumes you have a DISK II, but only the loading methods are different when using a cassette system (and the proper loading method has just been described). All explanations regarding the song data format are the same for any system. (Note: when using Applesoft, the word LOMEM in this paragraph refers to the start-of-program pointer.)

When using a system with a DISK II, you should change the PERFORM program into a binary file. Since you will probably want to still use the name PERFORM, you will have to delete the original PERFORM program since two programs cannot have the same name. To be on the safe side, you should begin by saving the original PERFORM program on some disk for possible future use. (Be sure you just LOAD PERFORM and then SAVE PERFORM on another disk. Do not RUN it or it will not be properly saved.) To begin, type INT (FP on Applesoft systems). Now load the PERFORM program (from cassette tape or disk). If you loaded it from disk, and wish to have the binary version of PERFORM on the same disk, you must DELETE PERFORM. Now RUN the program. Then type BSAVE PERFORM,A2050,L713 and press return. A binary file version of PERFORM will be saved on the disk. To finish, type INT (FP on Applesoft systems).

To copy the binary version of PERFORM to another disk, type BLOAD PERFORM,A2050 to load it, and then BSAVE PERFORM,A2050,L713 to save it on the desired disk.

If you wish to play an ENTRY-created song from your own BASIC program, it will first be necessary to convert the song into a binary file so your program can load it. In order to play a song, its data must be initialized to have the correct SLOT setting for your system. The easiest way to do this is to run a properly configured ENTRY program (see the ENTRY section), load the song and play it, then save the song back on disk. ENTRY's PLAY command will configure the song. (Note: you must remember to SAVE the configured song back on disk, or

the disk copy of the song will not be configured.) Once you have done this, you are ready to convert the song into a binary file. (Note that it will be necessary to reconfigure the song if you change the slot location of your synthesizer.) The following Integer BASIC program converts songs into binary files. Type it in and save it. Note that "d" means to type control D.

```
10 POKE 76,0 : POKE 77,124 : DIM A$(40) : INPUT "SONG NAME?";A$
20 PRINT "dLOADM:";A$ : A=PEEK(202)+PEEK(203)*256
30 PRINT "dBSAVE";A$;"A";A;"L";31744-A : PRINT "LENGTH: ";31744-A
40 PRINT "dINT"
```

Note: this program requires 48K. On 32K systems, change the 124 to an 80 and the two 31744's to 20480's. Songs entered on a 48K system with MAXFILES less than 3 (or on a cassette based system) may be too large to convert on a 32K system.

If you do not have Integer BASIC, use the following Applesoft version instead. Type it in and save it. Note that "d" means to type control D.

```
10 POKE 76,PEEK(115) : POKE 77,PEEK(116) : POKE 217,0
20 HIMEM:30000 : INPUT "SONG NAME?";A$
30 POKE PEEK(54)+PEEK(55)*256+3065,0
40 PRINT "dLOADM:";A$ : A=PEEK(202)+PEEK(203)*256
50 L=PEEK(76)+PEEK(77)*256-A : PRINT "dBSAVE";A$;"A";A;"L";L
60 PRINT "LENGTH: ";L : PRINT "dFP"
```

To use either program, begin by typing INT (FP for the Applesoft version). Then RUN the program. It will ask for a song name. Type in the name of the song to be converted (without the M:) and press return. The song will be converted and saved on your disk as a binary file with the same name as the song but without the M:. The conversion program also prints the length of the song in bytes. Although this length can be determined simply by BLOADing the song and looking at the DOS 3.2 file length locations (see your DOS manual), you may wish to write the length down since you will probably need to know it. To convert another song, follow the instructions above again. You can omit the initial INT (or FP), but you must load the program again to run it (or use RUN name) since the program self-destructs each time it is used.

AN EXAMPLE

Let's say you want to try this procedure with the sample song DIXIE BOOGIE. First, store a binary version of PERFORM as described above, and save the conversion program given above. Let's assume you named the conversion program CONVERT. Now, RUN ENTRY. (This assumes you have already configured ENTRY for

your system configuration as described in the ENTRY section.) LOAD:DIXIE BOOGIE, PLAY, and SAVE:DIXIE BOOGIE. Now type INTEGER to exit ENTRY. You are now ready to convert the song. Type INT (or FP). Type RUN CONVERT. It will ask for a song name. Type DIXIE BOOGIE and press return. The song will be converted and saved on your disk as DIXIE BOOGIE, and the length will be printed. (If you had another song to convert now, you would start with RUN CONVERT.) Now, the song can be played with PERFORM. To do this, begin with BLOAD PERFORM. Now type BLOAD DIXIE BOOGIE,A3059 and then POKE 2048,243 and POKE 2049,11. Type CALL 2050 to play the song. Note that paddle 0 controls the playback speed. When playback is finished, you could play the song again just by typing CALL 2050.

What are the mystic pokes for? Locations 2048 and 2049 must be set to the starting memory address of the song data. We loaded the song at 3059. Note that $11*256+243$ (11 and 243 being the numbers we poked) is 3059, the starting address. 3059 just happens to be the first byte of memory available after PERFORM, which uses locations 2048 through 3058.

With a few precautions, you could have had a BASIC program do the BLOADs, POKEs, and CALL. The only other detail is that in this example we used ENTRY to initialize the synthesizer (when DIXIE BOOGIE was configured), and for general purpose BASIC programs you would probably want to have your program initialize the synthesizer. If you were writing a program to be used on other people's computers, you would probably want to have your program configure the song data, too.

A FEW PRECAUTIONS

When using PERFORM from a BASIC program, you will have to find a place to put the song data. You will also have to keep BASIC from erasing PERFORM.

WITH INTEGER BASIC

When using Integer BASIC, the easiest place to put the song data is right after PERFORM. (Starting at 3059 decimal.) LOMEM can be moved up to keep BASIC from erasing either PERFORM or the song data. First, figure out where the song data will end. You will need to know the length of the longest song you plan to BLOAD, or the sum of the lengths of the longest songs you plan to have in memory at the same time. Take this length and add the starting address (3059). This is what LOMEM must be. You can either set LOMEM using a LOMEM command, or you can have your program set LOMEM. It is probably best to have your program do it so you won't forget, and so others can use it. The LOMEM command also changes a value Apple calls CM, so your program must change it too. To do all this, find out what $LOMEM \text{ MOD } 256$ and $LOMEM/256$ are (for the new LOMEM, of course). For example, if your longest song is less than 2048 bytes, LOMEM could

be $3059+2048$ which is 5107. $5107 \text{ MOD } 256$ is 243 and $5107/256$ is 19. To have your program change LOMEM and CM to these values, make the first statements `POKE 74,243 : POKE 204,243 : POKE 75,19 : POKE 205,19`. These four pokes must be the first statements in your program, or at least be before any variables are used. After these, you can BLOAD PERFORM by using `PRINT "dBLOAD PERFORM"` where the "d" is a control D. You can load a song using `PRINT "dBLOAD song name,A3059"` where "song name" is the name of the song to be loaded. If you wish to load a second song, change the 3059 after the A to a value which is 3059 plus the length of the first song or greater. Similarly, a third song can be loaded with an A value of 3059 plus the combined lengths of all previously loaded songs (or greater). Loading several songs lets you do a lot of disk reading at the beginning of the program (or any time before playback is needed) and then play any of the loaded songs at any time without delay. On the other hand, you may wish to just load a song, play it, then load another song and play it. This requires less memory, and it splits up the disk reading time. (When reading one song at a time, you only need enough memory to hold the longest song, and you BLOAD each song at the same address.) To play any song, you will need its starting address. This is the number after the A in the BLOAD command. The address MOD 256 must be poked at 2048, and the address/256 must be poked at 2049. Then a `CALL 2050` is used to play the song. If you load another song at the same address, you don't need to poke the starting address again. However, if you've loaded several songs, you will need to poke the starting address of the desired song before using `CALL 2050` to play it.

You can, of course, locate your song data any place it won't be erased. You must still move LOMEM up to at least 3059 to keep Integer BASIC from erasing PERFORM.

WITH APPLESOFT BASIC

Applesoft's memory organization is very crude, and thus more awkward preparations (than with Integer BASIC) are required. To begin with, your program should start with several REM statements. They should be line numbers 1 through 5. Line 1 should be typed in as `1REMXXXX...` with no spaces and with enough X's to completely fill 6 lines on the Apple's 40 column display. (There will be 235 X's. Don't type the ... of course.) Lines 2 through 5 must start as `2REMXXXX...` `3REMXXXX...` etc. These REM statements provide enough room for the PERFORM program. The next lines in your program should change Applesoft's start-of-program pointer to eliminate the REM's (while keeping enough room available for PERFORM). This is done with the statements `POKE 103,182 : POKE 104,12`. This is all you need to keep Applesoft from erasing PERFORM. Now an area of memory for the song(s) to be played is needed. The easiest area to use is the memory below the DOS system (below HIMEM). You will need to know the length of the longest song you plan to BLOAD, or the sum of the lengths of the longest songs you plan to have in memory at the same time. Let's call this number "length". Before your

program uses any variables, you should have this statement to reserve an area of memory (of length "length") for the song(s): `IF PEEK(2050)<>138 THEN HIMEM:PEEK(115)+PEEK(116)*256-length`. After this, you can use `PRINT "dBLOAD PERFORM"` to read in the PERFORM program (remember that "d" means to type control D). Now you should set a variable which indicates the address of this memory area. This is done with the statement `A=PEEK(115)+PEEK(116)*256`. You can load a song using `PRINT "dBLOAD song name,A";A` where "song name" is the name of the song to be loaded. If you wish to load a second song, use `PRINT "dBLOAD song name,A";A+L` where "song name" is the name of the second song, and L is the length of the first song. Similarly, a third song can be loaded using a value for L which is the combined lengths of all previously loaded songs. Loading several songs lets you do a lot of disk reading at the beginning of the program (or any time before playback is needed) and then play any of the loaded songs at any time without delay. On the other hand, you may wish to just load a song, play it, then load another song and play it. This requires less memory, and it splits up the disk reading time. (When reading one song at a time, you only need enough memory to hold the longest song, and you BLOAD each song at the same address.) To play any song, you will need to poke its starting address at 2048 and 2049. The starting address is the value A (or A+L) in the BLOAD command. Use `POKE 2048,A-INT(A/256)*256 : POKE 2049,A/256`. Then a `CALL 2050` is used to play the song. If you load another song at the same address, you don't need to poke the starting address again. However, if you've loaded several songs, you will need to poke the starting address of the desired song before using `CALL 2050` to play it.

You can, of course, locate your song data any place it won't be erased. You must still use the REM statements and the `POKE 103,182 : POKE 104,12` to keep Applesoft from erasing PERFORM.

CAUTION: when you run your Applesoft program, the REM statements will disappear. This will present no problems unless you save the program while the REM statements are gone. If you do, then sometime later (when PERFORM is no longer in memory) you may run the program and the first few lines would disappear, possibly causing bizzare listings (due to partial lines) and really odd RUNs after the first one. To repair this problem, just load the missing REM version from the disk and type in the REMs. To avoid having this problem occur, begin any session of correction by loading the program, running it to make the REMs disappear, then loading it again to bring the REMs back; this time the REMs will not disappear when you run the program since the start-of-program pointer has already been changed.

SYNTHESIZER INITIALIZATION

If you have a line which sets SLOT, like the one in ENTRY or PLAY, you can use this variable in a synthesizer initialization routine. Generally, any program which uses the synthesizer should have this initialization routine near the beginning. It is the same for either Integer BASIC or Applesoft.

```
PN=SLOT*16-16256
FOR P=PN TO PN+2
POKE P,159 : POKE P,191 : POKE P,223 : POKE P,255 : POKE P,231
NEXT P
```

SONG CONFIGURATION

Unless you can configure each song for your particular system (using ENTRY, as previously described) and can count on your program being used only on your system, you will need a song configuration routine. This routine uses SLOT, as does the synthesizer initialization routine (above). It also needs the variable A set to the starting address of the song to be configured.

The Integer BASIC version looks like this: (note: the song must not occupy address 32768)

```
FOR B=1 TO PEEK(A)
PNTR=PEEK(B+B+A-1)+PEEK(B+B+A)*256+A
CHAN=PEEK(PNTR+2) MOD 4 : NAHC=PEEK(PNTR+2) MOD 16 / 4
IF NAHC THEN NAHC=3-NAHC
POKE PNTR+1,SLOT*16+CHAN*4+NAHC
NEXT B
```

The Applesoft version looks like this:

```
FOR B=1 TO PEEK(A)
PNTR=PEEK(B+B+A-1)+PEEK(B+B+A)*256+A
CHAN=PEEK(PNTR+2)-INT(PEEK(PNTR+2)/4)*4
NAHC=INT((PEEK(PNTR+2)-INT(PEEK(PNTR+2)/16)*16)/4)
IF NAHC THEN NAHC=3-NAHC
POKE PNTR+1,SLOT*16+CHAN*4+NAHC
NEXT B
```

When using either version, you might wish to add POKE 2048,A MOD 256 : POKE 2049,A/256 : CALL 2050 : RETURN (which is POKE 2048,A-INT(A/256)*256 : POKE 2049,A/256 : CALL 2050 : RETURN in Applesoft) to the end in order to create a subroutine which can be GOSUBed in order to configure and play the song at address A.

READING THE "SUGGESTED SPEED"

Assuming the song was just loaded using PRINT "dLOAD song name,A";A the suggested speed from an ENTRY-created song can be read into the variable S with the following statement: S=PEEK(PEEK(-21920)+PEEK(-21919)*256+A-161). Note that the entire song must be located below memory address 32768 when using Integer BASIC. The -21920 is for a 48K system and must be -38304 on a 32K system. Likewise, the -21919 must be -38303 on a 32K system. In either case, Apple's DOS 3.2 must be used.

TEMPO CONTROL

If you wish to use a different paddle than Paddle 0 to control the playback speed, you must POKE 2346,n where n is the paddle number plus 100. (For a fixed playback speed, you may wish to install a 150K ohm 1/4 watt resistor at the game paddle connector between the +5 and PDL3 pins; then select paddle 3 for playback control as just described. Paddle 3 is an ideal choice since there is no switch input for this paddle, which may prohibit use of a real paddle.)

A SAMPLE SESSION

The following sample session is for a 48K system with Integer BASIC and Apple's DOS 3.2. The changes necessary for a 32K system or for Applesoft (or both) have already been discussed above. It is assumed that the PERFORM program and the M:DIXIE BOOGIE song, as provided with the synthesizer, are already on disk.

```
>LOAD PERFORM
>DELETE PERFORM
>RUN
PERFORM ALF PRODUCTS INC.
```

CONVERT PERFORM TO A BINARY FILE

```
>BSAVE PERFORM,A2050,L713
>INT
```

```
>10 POKE 76,0 : POKE 77,124 : DIM A$(40) : INPUT "SONG NAME?",A$
>20 PRINT "dLOADM:";A$ : A=PEEK(202)+PEEK(203)*256
>30 PRINT "dBSAVE";A$;"A";A";L";31744-A : PRINT "LENGTH: ";31744-A
>40 PRINT "dINT"
```

**SAVE
CONVERT
PROGRAM**

```
>SAVE CONVERT
>RUN
SONG NAME?DIXIE BOOGIE
```

**CONVERT
DIXIE BOOGIE
TO BINARY**

```
LENGTH: 3298
```

```
>INT
```

```

>6 POKE 74,0 : POKE 204,0 : POKE 75,64 : POKE 205,64
>10 SLOT=4 (CHANGE AS REQUIRED)
>30 PRINT "dLOAD PERFORM" : DIM A$(40)
>40 PN=SLOT*16-16256
>50 FOR P=PN TO PN+2
>60 POKE P,159 : POKE P,191 : POKE P,223 : POKE P,255 : POKE P,231
>70 NEXT P
>80 INPUT "SONG NAME?",A$ : A=3059 : PRINT "dLOAD";A$;"A";A
>90 S=PEEK(PEEK(-21920)+PEEK(-21919)*256+A-161)
>100 IF NOT S THEN 180
>160 PRINT "SUGGESTED SPEED: ";S
>170 PRINT PDL(0);" "; : TAB 1 : IF PEEK(-16287)<128 THEN 170
>180 FOR B=1 TO PEEK(A)
>190 PNTR=PEEK(B+B+A-1)+PEEK(B+B+A)*256+A
>200 CHAN=PEEK(PNTR+2) MOD 4 : NAHC=PEEK(PNTR+2) MOD 16 / 4
>210 IF NAHC THEN NAHC=3-NAHC
>215 POKE PNTR+1,SLOT*16+CHAN*4+NAHC
>220 NEXT B : POKE 2048,A MOD 256 : POKE 2049,A/256
>230 CALL 2050 : GOTO 80
>SAVE YALP
>RUN
SONG NAME?DIXIE BOOGIE

```

**PROTECT PERFORM
AND SONG AREA**

LOAD PERFORM

INITIALIZE SYNTHESIZER

READ SONG

READ SUGGESTED SPEED

CONFIGURE SONG

PLAY THE SONG

CREATION OF A SIMPLE PROGRAM

SUGGESTED SPEED: 160

(etc.) **(SONG PLAYS WHEN BUTTON IS PRESSED)**

TECHNICAL

PERFORM operates on one to nine sequences of commands stored in memory. Each sequence of commands indicates the sounds for one channel. All the sequences will appear to be executed at the same time by PERFORM. There are three types of commands which may be used. One type is used to control the execution of the commands. Another type is used to set parameters for future use. The remaining type of command is used to wait or to produce a new pitch and wait. During the time "waited", PERFORM will automatically program volume settings which create the selected envelopes. Envelope production is explained in the ENTRY section and in the block diagram at the end of this section.

All commands for PERFORM are three bytes long. (Each byte is an integer from 0 to 255.) The first byte always indicates the particular command desired, and the second and third bytes indicate a parameter for use by that command. When the parameter is a two-byte integer (0 to 65535), the low byte (value MOD 256) is given as the second byte of the command and the high byte (value/256) is given as the third byte. The various commands available are described below.

TYPE A COMMANDS

The first type of command is used to control execution. They are CHANNEL NUMBER, CALL, RETURN, STOP, and END.

CHANNEL NUMBER

The CHANNEL NUMBER command is used to indicate the slot and channel number to be programmed. The second byte should be 16 times the expansion slot number plus the stereo position code (0=left, 1=right, 2=middle) plus 4 times the channel number (0-2) within the stereo position. Although PERFORM does not use the third byte, it should be used to indicate stereo positioning. Its most significant four bits indicate stereo positioning for performance with two AMS units (meaningless for the Apple Music], see the 11-1-6 AMS manual), and the least significant four bits indicate stereo positioning for the Apple Music]. In this lower half byte, the two most significant bits indicate a stereo position code (0=left, 1=middle, 2=right). The two least significant bits indicate the channel number within the stereo position (0 to 2). The first command in each part must be a CHANNEL NUMBER command. ENTRY compatible songs may have only one CHANNEL NUMBER command per part.

CALL

The CALL command is used to perform a subroutine call. The second and third bytes indicate the relative address of the subroutine. During playback, the commands in the subroutine will be executed, and then PERFORM will continue in

the usual fashion with the commands following the CALL.

RETURN

The RETURN command marks the end of a subroutine, and causes PERFORM to continue at the commands following the CALL. The second and third bytes must be the same as the second and third bytes of the CALL command. ENTRY compatible songs may have only one RETURN command per subroutine.

STOP

The STOP command indicates the end of one part's (or channel's) commands. The envelope generator will continue to operate after a STOP command if no other channel has encountered an END command. The second and third bytes are not used and should be set to 0. All parts except the last one should end with a STOP command. ENTRY compatible songs may have only one STOP command per part.

END

The END command is used to terminate PERFORM and return to the calling program. The last part should end with an END command rather than a STOP command. Further, the END command should be positioned as the last command in all the data (in ENTRY compatible songs, this is followed by the "suggested speed" byte and the 160 title bytes). Envelope production does not continue once any part executes an END command. The second and third bytes are not used, and should be set to 0.

TYPE B COMMANDS

The second type of command is used to set parameters. They are TRANSPOSE, GAP SIZE, ATTACK RATE, DECAY RATE, VOLUME LEVEL, SUSTAIN LEVEL, RELEASE RATE, and FUZZ.

TRANSPOSE

The TRANSPOSE command is used to add or subtract a constant from all following pitch values (until a new TRANSPOSE value is programmed). The second byte indicates the amount to add or subtract. 0 to 127 will add a value of 0 to 127. 128 to 255 will subtract a value of 128 to 1. Since the values are in quarter-steps, adding a value of 24 will raise the pitch by one octave. The third byte is the pitch mask byte. All following pitch values are ANDed with the pitch mask byte (before the second byte transpose value is added or subtracted). This byte is normally set to 255. ENTRY compatible songs use a value of 254 to allow sharp/flat display selection with the least significant pitch bit. If the resultant pitch number is less than 30, it will be raised by multiples of 24 until it is 30 or greater.

FUZZ

The FUZZ command is used to select "white noise" (fuzz) mode or normal mode. Since the "pitch" of the white noise is always controlled by channel 2 (of a given stereo position), FUZZ should be used only on parts operating on channel 2. This command is identified by the first two bytes, unlike all other commands which are identified by the first byte alone. The third byte must be 0 for "normal" or 96 for "fuzz".

GAP SIZE

The GAP SIZE command is used to control the release stage of envelope production. When the number of time periods remaining to wait (during a "wait") equals the GAP SIZE value, the envelope parameters will automatically be changed. The RELEASE RATE value will be copied into the CURRENT DECAY RATE, and a 0 will be written into the DESIRED LOUDNESS and the CURRENT SUSTAIN LEVEL. This causes the CURRENT LOUDNESS (and therefore the volume) of the channel to drop to 0 at the RELEASE RATE. The second and third bytes indicate the new GAP SIZE. When a release stage is not desired, the GAP SIZE should be set to 65535 (255,255).

ATTACK RATE, DECAY RATE, VOLUME LEVEL, SUSTAIN LEVEL, RELEASE RATE

These commands are used to set envelope parameters. The second and third bytes indicate the new value.

TYPE C COMMANDS

The third type of command is used to wait or to produce a new pitch and wait. The second and third bytes indicate the number of time periods to wait before continuing with the next command. During this wait, the envelope generator program in PERFORM will update the envelope parameters and reprogram the volume once each time period. These commands are PITCH and REST.

PITCH

There are 192 PITCH commands with command numbers from 0 to 191. The range of the Apple Music]] is limited to values from 30 to 174 (other values are allowed for compatibility with the Apple Music Synthesizer (AMS)). (Note: see TRANSPOSE for additional information.) The command number indicates which pitch is to be produced, subject to modification by the two TRANSPOSE parameters. The resultant number specifies the pitch to be programmed into the synthesizer. Pitch specification is in quarter-steps, with 0 being A natural at 27.5 Hz. There are 24 quarter-steps per octave. Thus, 24 is A natural at 55 Hz. (Note: the synthesizer cannot play pitches with values below 30; such pitches will be transposed up by one or more octaves.) Note that in ENTRY compatible songs, the least significant bit of the PITCH command number indicates whether sharp or flat should be displayed, and is masked off during playback (see TRANPOSE). The

PITCH command also changes certain envelope parameters. The DECAY RATE is copied into the CURRENT DECAY RATE, the VOLUME LEVEL is copied into the DESIRED LOUDNESS, and the SUSTAIN LEVEL is copied into the CURRENT SUSTAIN LEVEL (see the block diagram at the end of this section).

REST

The REST command causes the RELEASE RATE to be copied into the CURRENT DECAY RATE, and a Ø to be written into the DESIRED LOUDNESS and the CURRENT SUSTAIN LEVEL. This causes the release portion of the envelope to begin. (Note: this is the same process as caused by the time remaining equaling the GAP SIZE, see the GAP SIZE command.)

SONG DATA

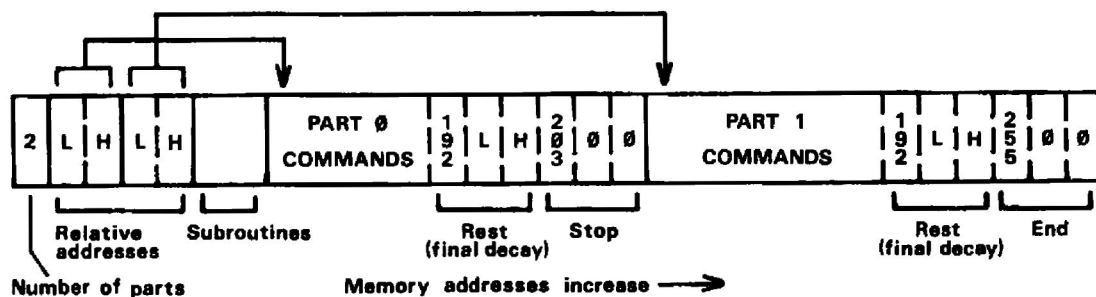
RELATIVE ADDRESSES

All relative addresses used in PERFORM (for example, the second and third bytes of a CALL command) must be two-byte integers stored low byte first. The value stored must be the actual memory address minus the starting address of the song data.

START OF DATA

The first byte (stored at the starting address) must be the number of "parts" of data. This must be an integer from 1 to 9. The following 2 to 18 bytes must be the relative address of the first command of each part. Following these bytes the subroutines (if any) are stored, and then the first part's commands, the second's, and so forth. See the diagram below.

Two Part Song Data



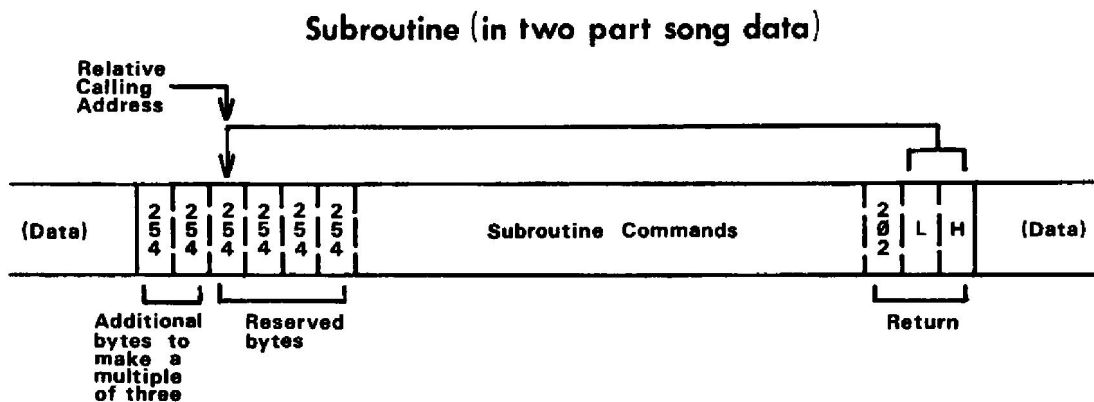
PART DATA

In each part, the three-byte commands are stored one after another. Each part must begin with a CHANNEL NUMBER command, and end with a STOP command (except the last part must end with an END command). See the diagram above. Although a part may contain more than one CHANNEL command, to do so would be incompatible with ENTRY and with the "song configuration" routine given earlier in this

section.

SUBROUTINE DATA

The relative calling address to a subroutine must point to several bytes of reserved storage which precede the first command of the subroutine. There must be two times as many reserved bytes as the number of parts. These reserved bytes must be preceded by at least 1 additional byte(s), and the number of additional bytes plus the number of reserved bytes must be evenly divisible by 3. See the diagram below.



Note that the calling address must point to the first of the reserved bytes, not to the additional bytes nor to the first command in the subroutine. The additional bytes must be stored as 254's, and the reserved bytes should be set to 254 also. When a CALL command is executed during playback, the address of the first command after the CALL (that is, the return address) is stored in two of the reserved bytes. (PERFORM assigns a different pair of bytes for each part. This allows several parts to call the subroutine at once.) The RETURN command at the end of the subroutine causes the address of the next-command-to-be-interpreted to be read from the correct pair of reserved bytes, thus causing a "return". Note that although a subroutine may contain more than one RETURN command (or a RETURN command to a different subroutine), to do so would be incompatible with ENTRY.

TEMPO COMMAND

The TEMPO command is ignored, but is allowed for compatibility with the 10-5-16 Apple Music Synthesizer (AMS). Information on the TEMPO command can be found in the AMS owner's manual, order number 11-1-6.

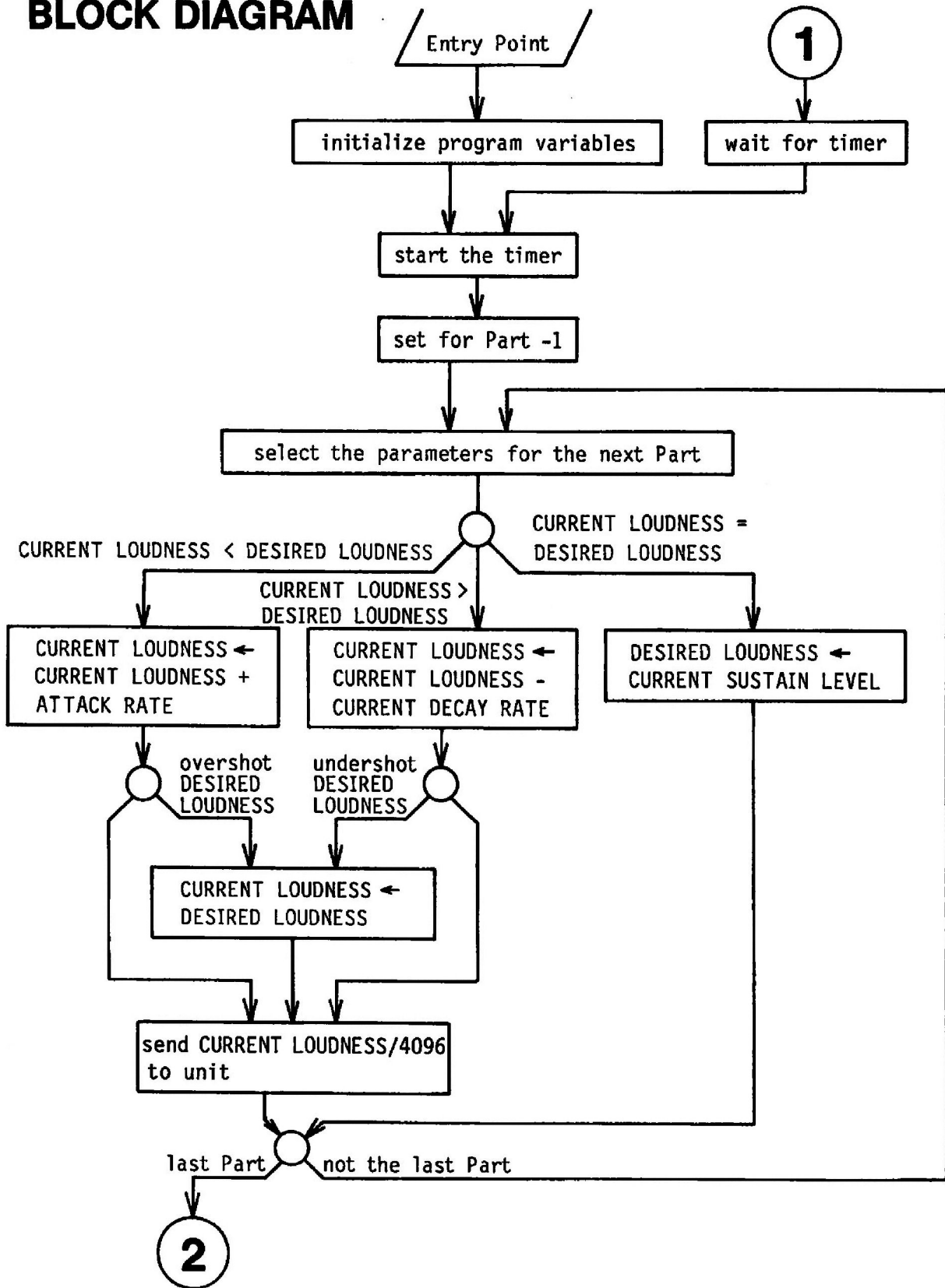
TEMPORARIES

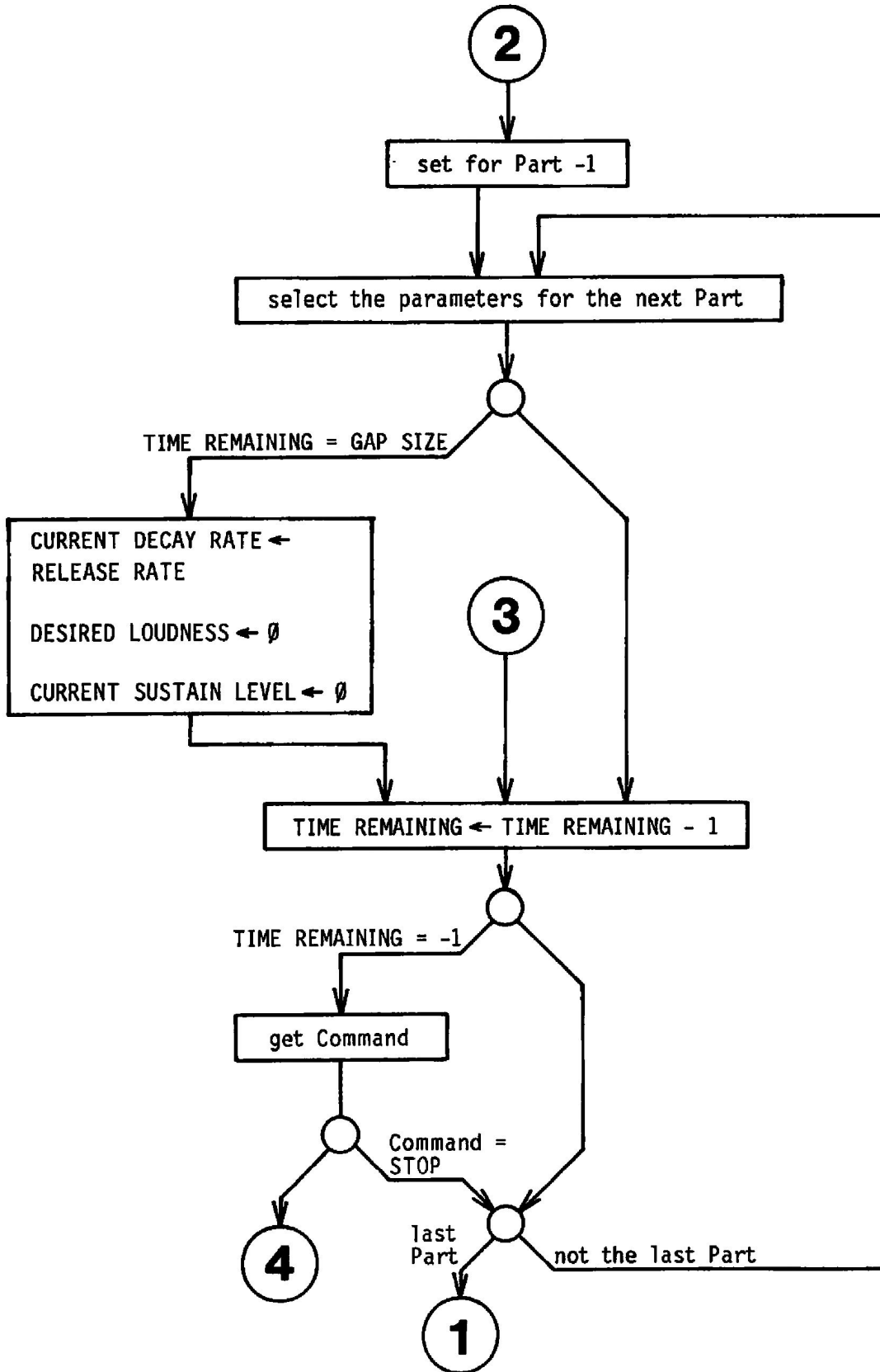
PERFORM uses locations 0-19 (hex) (6-C and DD-EF for the Applesoft version) for storage of temporary values during execution.

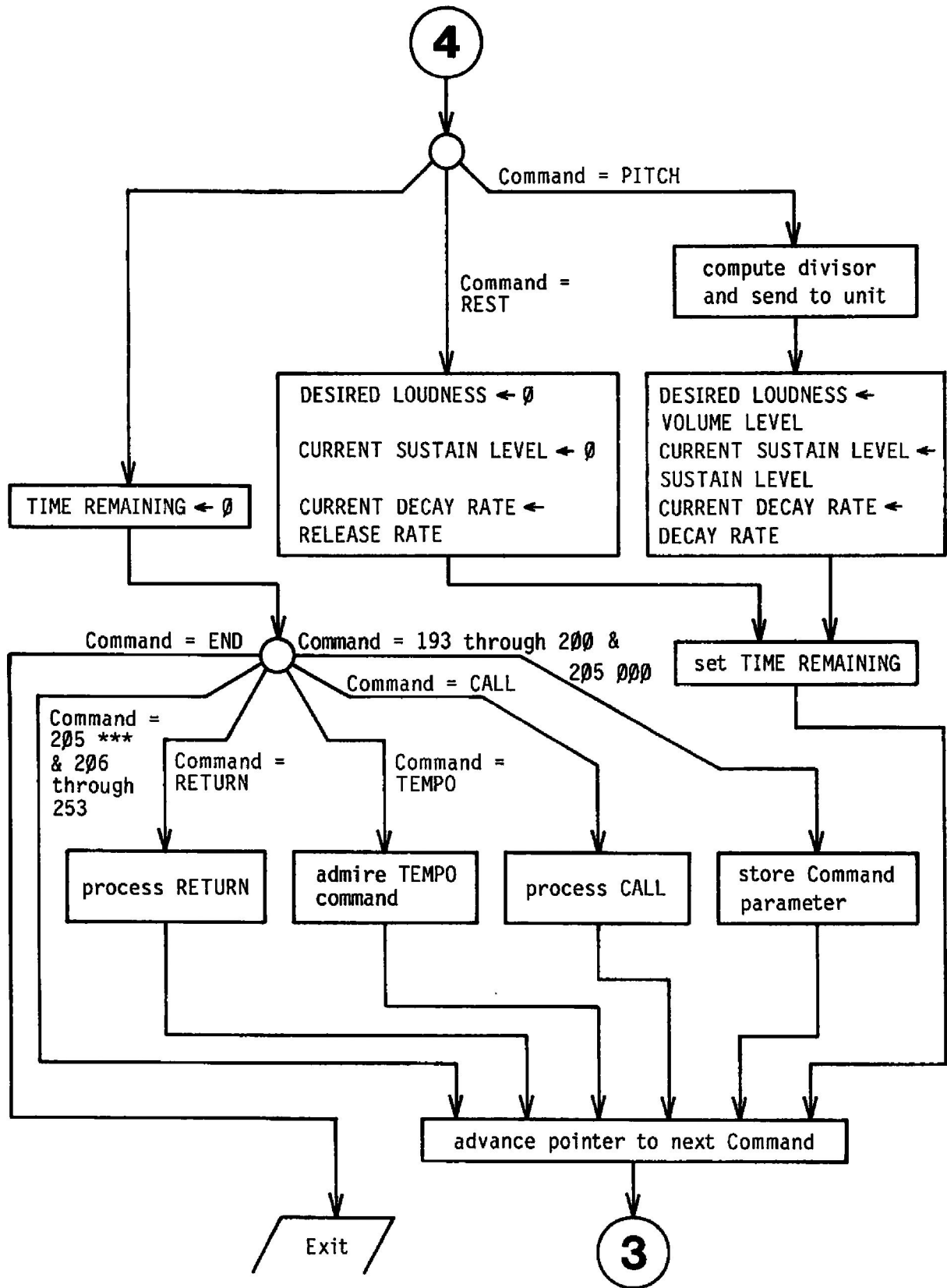
COMMAND NUMBERS

HEX	DECIMAL	COMMAND
0-BF	0-191	PITCH
C0	192	REST
C1	193	GAP SIZE
C2	194	TRANSPOSE
C3	195	ATTACK RATE
C4	196	DECAY RATE
C5	197	VOLUME LEVEL
C6	198	SUSTAIN LEVEL
C7	199	RELEASE RATE
C8	200	CHANNEL NUMBER
C9	201	CALL
CA	202	RETURN
CB	203	STOP
CC	204	TEMPO
CD 00	205 000	FUZZ
CD **	205 ***	(where ** or *** are non-zero) no operation
CE-FD	206-253	no operation
FE	254	preceeds subroutines, treated as END if found
FF	255	END

BLOCK DIAGRAM







6

PROGRAMMING

BARE HANDED

The Apple Music][is programmed by means of 3 "ports". Each port controls three channels (numbered 0-2) in a particular stereo position. Each port has been assigned a particular memory address, and information can be sent to a port by writing a byte (an integer from 0 to 255) to that memory address (using 6502 Assembly Language or BASIC's POKE). Reading from these memory addresses does not affect the synthesizer. The ports are numbered from 0 to 2. The memory address of each port is calculated by the formula $SLOT*16-16256+P$ where SLOT is the expansion slot number used by the synthesizer and P is the stereo position number (0=left, 1=right, and 2=middle).

INITIALIZATION

Before use, the synthesizer should be initialized. Upon power on, the synthesizer will generally produce random tones at high volumes. Therefore, the volumes of all nine channels, plus the three noise channels, should be programmed to zero. Additionally a mode control byte must be sent to each stereo position. The mode control value is 231. Thus, a BASIC initialization program would appear as follows:

```
FOR P=0 TO 2
A=SLOT*16-16256+P
POKE A,159 : POKE A,191 : POKE A,223 : POKE A,255
POKE A,231
NEXT P
```

The first line of pokes zeros the volumes, and the second poke line sets the mode.

VOLUME

The volume levels of each of the three channels for each stereo position, and the volume levels of the white noise in each stereo position, can all be programmed independently. The formula $159+32*C-V$, where C is the channel number (0 to 3) and V is the volume (0 to 15) is used to program volume. Channel 3 is the white noise volume. For example, to set channel 2 to full volume (15), you would write $159+32*2-15$ which is 208. This could be done using `POKE SLOT*16-16256+P,208`.

The amount of attenuation (from full volume) is approximately $30-2*V$ dB for values of V from 1 to 15. $V=0$ is "off" (infinite attenuation).

FREQUENCY

The frequencies (itches) of each of the three channels for each stereo position can all be programmed independently. (Note: the pitch of channel 2 affects the

white noise output as well.) Any frequency $63920/D$ Hz, where D is an integer from 1 to 1023, can be selected. Two bytes must be written. The first byte is computed by the formula $128+32*C+D \text{ MOD } 16$, where C is the channel number (0 to 2) and D is the frequency divisor (1 to 1023). Using INT instead of MOD, the formula is $128+32*C+D-\text{INT}(D/16)*16$. The second byte is computed by the formula $D/16$, or $\text{INT}(D/16)$. Both bytes must be written at address $\text{SLOT}*16-16256+P$, and the first byte must be written first.

6502 PROGRAMMING

When programming in 6502 Assembly (or Machine) Language, an additional consideration occurs. 18 cycles must be allowed to pass between each write. This is normally a concern only when programming the frequency, since then two bytes must be written in succession. The 18 cycle requirement is easily met by arranging calculations to occur between the two frequency byte writes, or by inserting NOPs.

DIVISOR CALCULATION

Pitches and volumes must increase (and decrease) exponentially to achieve an apparent linear increase (for humans). Exponential volume increases are automatically created by the exponential amplifiers in the volume control circuitry. Exponential pitch increases must be created by selecting divisors which result in exponentially higher (and lower) pitches.

The most common exponential pitch spacing is the equal tempered scale, which is similar to the piano scale. This scale is divided into "octaves" with 12 notes per octave (half tones) or 24 notes per octave (quarter tones) depending on the application. An octave is defined to mean that the frequency of a note is twice that of the same note in the next lower octave. The frequency, $F(N)$, of any particular note, N, in an octave is calculated by $F(N)=F(0)*(2 \uparrow (N/X))$ where X is the number of notes per octave, $F(0)$ is the frequency (pitch) of the lowest note in the octave (in Hz, or cycles per second), and N must be an integer from 0 to X-1. (Note: although written in standard BASIC format, the formulas here are not intended to be computed in BASIC without careful consideration of the accuracy required. Floating-point calculation should be used in any case.) The frequency, $F(N,Q)$, of any given note, N, in any given octave, Q, is calculated $F(N,Q)=F(N,0)*(2 \uparrow Q)$ where $F(N,0)$ is equivalent to $F(N)$ in the previous formula and Q is an integer. The lowest note on a piano has a frequency of 27.5 Hz (using standard A=440 Hz tuning). Thus the frequency, $F(N,Q)$, of any piano note is $F(N,Q)=27.5*(2 \uparrow (Q+N/12))$ Hz, where N is the note number from 0 to 11 and Q is the octave number from 0 to 7. (Note: pianos have no notes where N is greater than 3 if Q is 7. N=0 indicates an A natural pitch.) Therefore the desired divisors for piano notes are: $D(N,Q)=\text{INT}(63920/(27.5*(2 \uparrow (Q+N/12))))+0.5$. Note that the 12

can be replaced with a 24 (and the range of N extended to 0-23) to obtain quarter tones. It is usually convenient to calculate divisors using a small look-up table containing $D(N, \emptyset)$ and dividing by 2^Q and rounding. This is easily accomplished in Assembly Language by shifting the divisor right Q times (shifting in 0's) and then adding in the last bit shifted out in order to round.

Note that since divisors less than 1 or greater than $1/23$ cannot be used, the lowest notes on a piano cannot be programmed. The highest notes cannot be programmed since the tuning inaccuracy becomes very large at higher frequencies (smaller divisors).

7

TECHNICAL

PERFORM (INTEGER VERSION)

0000		10		* 10-5-1 PERFORM SUBROUTINE	
0000		20		*	
0000		30		* BY JOHN RIDGES	
0000		40		*	
0000		50		* ALF PRODUCTS INC.	
0000		60		*	
0000		70		ORG 0	
0000		80		* BASE PAGE USAGE	
0000		90	SPNTR	BSS 2	SONG DATA POINTER
0002		100	COUNT	BSS 1	PART COUNTER
0003		110	TEMP1	BSS 1	
0004		120	TEMP2	BSS 1	
0005		130	TEMP3	BSS 2	
0007		140	PARNUM	BSS 19	NUMBER OF PARTS
0008		150	PARPNT	EQU PARNUM+1	PART POINTERS
0800		160		ORG \$800	
0800		170		* SUBROUTINE PARAMETER	
0800		180	DPNTR	BSS 2	SONG DATA BEGINNING ADDRESS
0802		190		* SUBROUTINE ENTRY POINT	
0802	8A	200		TXA	SAVE X
0803	48	210		PHA	REGISTER
0804	AD 00 08	220		LDA DPNTR	SET SONG
0807	85 00	230		STA /SPNTR	DATA POINTER
0809	AD 01 08	240		LDA DPNTR+1	
080C	85 01	250		STA /SPNTR+1	
080E	A0 00	260		LDY #0	GET NUMBER
0810	B1 00	270		LDA (SPNTR),Y	OF PARTS
0812	0A	280		ASL A	
0813	85 07	290		STA /PARNUM	
0815	A2 00	300		LDX #0	SET UP
0817	C8	310	CPYADR	INY	PART POINTERS
0818	B1 00	320		LDA (SPNTR),Y	
081A	18	330		CLC	
081B	65 00	340		ADC /SPNTR	
081D	95 08	350		STA /PARPNT,X	
081F	E8	360		INX	
0820	C8	370		INY	
0821	B1 00	380		LDA (SPNTR),Y	
0823	65 01	390		ADC /SPNTR+1	
0825	95 08	400		STA /PARPNT,X	
0827	E8	410		INX	
0828	E4 07	420		CPX /PARNUM	
082A	D0 EB	430		BNE CPYADR	
082C	46 07	440		LSR /PARNUM	
082E	A2 F3	450		LDX #ASIZE*9	CLEAR
0830	A9 00	460		LDA #0	PARAMETER AREA
0832	9D FF 0A	470	CLEAR	STA TIME-1,X	
0835	CA	480		DEX	
0836	D0 FA	490		BNE CLEAR	
0838		500		* MAIN EXECUTION LOOP	
0838	A5 07	510		LDA /PARNUM	SET UP

083A	85	02	520		STA /COUNT	PART COUNTER
083C	A2	00	530	MAIN	LDX #0	
083E	8D	70	540		STA \$C070	START TIMER
0841			550			
0841	BD	12	0B		LDA LOUDNS,X	CHECK CL (CURRENT LOUDNESS)
0844	38		570		SEC	AND DL (DESIRED LOUDNESS)
0845	FD	16	0B		SBC DESIRE,X	
0848	85	03	590		STA /TEMP1	
084A	BD	13	0B		LDA LOUDNS+1,X	
084D	FD	17	0B		SBC DESIRE+1,X	
0850	90	12	620		BCC UPLD	BRANCH IF CL<DL
0852	05	03	630		ORA /TEMP1	
0854	D0	31	640		BNE DWNLD	BRANCH IF CL>DL
0856	BD	18	0B		LDA CURSUS,X	CL=DL
0859	9D	16	0B		STA DESIRE,X	DL:=CURRENT SUSTAIN LEVEL
085C	BD	19	0B		LDA CURSUS+1,X	
085F	9D	17	0B		STA DESIRE+1,X	
0862	B0	70	690		BCS NEXT	
0864	BD	12	0B	UPLD	LDA LOUDNS,X	CL:=CL+ATTACK RATE
0867	7D	06	0B		ADC ATTACK,X	
086A	9D	12	0B		STA LOUDNS,X	
086D	BD	13	0B		LDA LOUDNS+1,X	
0870	7D	07	0B		ADC ATTACK+1,X	
0873	9D	13	0B		STA LOUDNS+1,X	
0876	B0	31	760		BCS ETHERE	BRANCH IF OVERSHOT DL
0878	A8		770		TAY	COMPARE CL AND DL
0879	BD	12	0B		LDA LOUDNS,X	
087C	DD	16	0B		CMP DESIRE,X	
087F	98		800		TYA	
0880	FD	17	0B		SBC DESIRE+1,X	
0883	90	3C	820		BCC SENDE	DON'T BRANCH IF OVERSHOT DL
0885	B0	22	830		BCS ETHERE	
0887	BD	12	0B	DWNLD	LDA LOUDNS,X	CL:=CL-CURRENT DECAY RATE
088A	FD	14	0B		SBC DOWN,X	
088D	9D	12	0B		STA LOUDNS,X	
0890	BD	13	0B		LDA LOUDNS+1,X	
0893	FD	15	0B		SBC DOWN+1,X	
0896	9D	13	0B		STA LOUDNS+1,X	
0899	90	0E	900		BCC ETHERE	BRANCH IF UNDERSHOT DL
089B	BD	16	0B		LDA DESIRE,X	COMPARE CL AND DL
089E	DD	12	0B		CMP LOUDNS,X	
08A1	BD	17	0B		LDA DESIRE+1,X	
08A4	FD	13	0B		SBC LOUDNS+1,X	
08A7	90	18	950		BCC SENDE	DON'T BRANCH IF UNDERSHOT DL
08A9	BD	16	0B	ETHERE	LDA DESIRE,X	CL:=DL
08AC	9D	12	0B		STA LOUDNS,X	
08AF	BD	17	0B		LDA DESIRE+1,X	
08B2	9D	13	0B		STA LOUDNS+1,X	
08B5	BD	18	0B		LDA CURSUS,X	DL:=CURRENT SUSTAIN LEVEL
08B8	9D	16	0B		STA DESIRE,X	
08BB	BD	19	0B		LDA CURSUS+1,X	
08BE	9D	17	0B		STA DESIRE+1,X	
08C1	BC	10	0B	SENDE	LDY CHAN,X	SEND LOUDNESS
08C4	BD	13	0B		LDA LOUDNS+1,X	TO UNIT

08C7	4A		1060		LSR A	
08C8	4A		1070		LSR A	
08C9	4A		1080		LSR A	
08CA	4A		1090		LSR A	
08CB	5D	11	0B	1100	EOR CHAN+1,X	
08CE	1D	1A	0B	1110	ORA PERCUS,X	
08D1	99	80	C0	1120	STA \$C080,Y	
08D4	8A			1130	NEXTE TXA	REPEAT FOR
08D5	18			1140	CLC	NEXT PART
08D6	69	1B		1150	ADC #ASIZE	
08D8	AA			1160	TAX	
08D9	C6	02		1170	DEC /COUNT	
08DB	F0	03		1180	BEQ CONT1	
08DD	4C	41	08	1190	JMP ENVEL	
08E0	A2	00		1200	CONT1 LDX #0	INITIALIZE PART COUNTER
08E2				1210	* NOTE DURATION SECTION	
08E2	BD	00	0B	1220	LENGTH LDA TIME,X	COMPARE TIME REMAINING
08E5	DD	02	0B	1230	CMP GAP,X	AND GAP SIZE
08E8	D0	22		1240	BNE DECR	BRANCH IF UNEQUAL
08EA	BD	01	0B	1250	LDA TIME+1,X	
08ED	DD	03	0B	1260	CMP GAP+1,X	
08F0	D0	1A		1270	BNE DECR	BRANCH IF UNEQUAL
08F2	BD	0E	0B	1280	LDA RELEAS,X	EQUAL; START NOTE
08F5	9D	14	0B	1290	STA DOWN,X	RELEASE
08F8	BD	0F	0B	1300	LDA RELEAS+1,X	CURRENT DECAY RATE:=
08FB	9D	15	0B	1310	STA DOWN+1,X	RELEASE RATE
08FE	A9	00		1320	LDA #0	DL:=0
0900	9D	16	0B	1330	STA DESIRE,X	CURRENT SUSTAIN LEVEL:=0
0903	9D	17	0B	1340	STA DESIRE+1,X	
0906	9D	18	0B	1350	STA CURSUS,X	
0909	9D	19	0B	1360	STA CURSUS+1,X	
090C	BD	00	0B	1370	DECR LDA TIME,X	DECREMENT TIME REMAINING
090F	D0	08		1380	BNE DECR2	
0911	BD	01	0B	1390	LDA TIME+1,X	
0914	F0	1B		1400	BEQ PROCES	BRANCH IF NO TIME LEFT
0916	DE	01	0B	1410	DECR1 DEC TIME+1,X	
0919	DE	00	0B	1420	DECR2 DEC TIME,X	
091C	8A			1430	NEXTL TXA	CONTINUE WITH
091D	18			1440	CLC	NEXT PART
091E	69	1B		1450	ADC #ASIZE	
0920	AA			1460	TAX	
0921	E6	02		1470	INC /COUNT	
0923	A5	02		1480	LDA /COUNT	
0925	C5	07		1490	CMP /PARNUM	
0927	D0	B9		1500	BNE LENGTH	
0929	2C	64	C0	1510	WAIT BIT \$C064	WAIT FOR TIMER
092C	30	FB		1520	BMI WAIT	
092E	4C	3C	08	1530	JMP MAIN	
0931				1540	* SONG DATA COMMAND PROCESSING SECTION	
0931	8A			1550	PROCES TXA	
0932	A8			1560	TAY	
0933	A5	02		1570	LDA /COUNT	
0935	0A			1580	ASL A	
0936	AA			1590	TAX	

0937	A1 08	1600	LDA (PARPNT,X)	GET COMMAND TYPE
0939	C9 CB	1610	CMP #203	
093B	D0 04	1620	BNE NOSTOP	BRANCH IF NOT "STOP"
093D		1630	* PROCESS STOP COMMAND	
093D	98	1640	TYA	DO NOTHING
093E	AA	1650	TAX	
093F	B0 D5	1660	BCS DECR1	
0941	F6 08	1670	NOSTOP INC /PARPNT,X	
0943	D0 02	1680	BNE NOCAR1	
0945	F6 09	1690	INC /PARPNT+1,X	
0947	C9 C0	1700	NOCAR1 CMP #192	
0949	B0 65	1710	BCS NPITCH	BRANCH IF NOT "PITCH"
094B		1720	* PROCESS PITCH COMMAND	
094B	39 05 0B	1730	AND TRANS+1,Y	
094E	79 04 0B	1740	ADC TRANS,Y	
0951	38	1750	SEC	
0952	E9 1E	1760	SBC #30	
0954	B0 04	1770	BCS PITCH2	MOVE THE LOWEST 30
0956	69 18	1780	PITCH1 ADC #24	PITCHS UP AN OCTAVE
0958	90 FC	1790	BCC PITCH1	OR TWO
095A	86 04	1800	PITCH2 STX /TEMP2	COMPUTE DIVISOR
095C	A2 00	1810	LDX #0	DIVIDE PITCH BY 24
095E	C9 18	1820	DIV CMP #24	
0960	90 05	1830	BCC DIV1	
0962	E9 18	1840	SBC #24	
0964	E8	1850	INX	
0965	D0 F7	1860	BNE DIV	
0967	86 03	1870	DIV1 STX /TEMP1	
0969	0A	1880	ASL A	LOOK UP
096A	AA	1890	TAX	SUB-OCTAVE DIVISOR
096B	BD 9C 0A	1900	LDA TABLE+1,X	
096E	85 05	1910	STA /TEMP3	
0970	BD 9B 0A	1920	LDA TABLE,X	
0973	A6 03	1930	LDX /TEMP1	
0975	CA	1940	OCTAVE DEX	DIVIDE DIVISOR
0976	30 06	1950	BMI ROUND	TO RIGHT OCTAVE
0978	46 05	1960	LSR /TEMP3	
097A	6A	1970	ROR A	
097B	4C 75 09	1980	JMP OCTAVE	
097E	69 08	1990	ROUND ADC #8	ROUND RESULT
0980	90 02	2000	BCC SENDP	
0982	E6 05	2010	INC /TEMP3	
0984	BE 10 0B	2020	SENDP LDX CHAN,Y	SEND PITCH TO UNIT
0987	09 0F	2030	ORA #F	
0989	4A	2040	LSR A	
098A	6A	2050	ROR A	
098B	6A	2060	ROR A	
098C	6A	2070	ROR A	
098D	39 11 0B	2080	AND CHAN+1,Y	
0990	9D 80 C0	2090	STA \$C080,X	
0993	84 03	2100	STY /TEMP1	
0995	EA	2110	NOP	
0996	EA	2120	NOP	
0997	EA	2130	NOP	

0998	EA	2140		NOP	
0999	A5 05	2150		LDA /TEMP3	
099B	9D 80 C0	2160		STA \$C080,X	
099E	A2 06	2170		LDX #6	START "ADSR" CYCLE
09A0	B9 08 0B	2180	CYCLE	LDA DECAY,Y	
09A3	99 14 0B	2190		STA DOWN,Y	
09A6	C8	2200		INY	
09A7	CA	2210		DEX	
09A8	D0 F6	2220		BNE CYCLE	
09AA	A6 04	2230		LDX /TEMP2	
09AC	A4 03	2240		LDY /TEMP1	STORE NOTE TIME
09AE	B0 22	2250		BCS STORD1	
09B0	D0 3B	2260	NPITCH	BNE NREST	BRANCH IF NOT "REST"
09B2		2270	* PROCESS	REST COMMAND	
09B2	B9 0E 0B	2280		LDA RELEAS,Y	DO A "RELEASE"
09B5	99 14 0B	2290		STA DOWN,Y	
09B8	B9 0F 0B	2300		LDA RELEAS+1,Y	
09BB	99 15 0B	2310		STA DOWN+1,Y	
09BE	A9 00	2320		LDA #0	
09C0	99 16 0B	2330		STA DESIRE,Y	
09C3	99 17 0B	2340		STA DESIRE+1,Y	
09C6	99 18 0B	2350		STA CURSUS,Y	
09C9	99 19 0B	2360		STA CURSUS+1,Y	
09CC	18	2370		CLC	
09CD	84 03	2380	STORD	STY /TEMP1	STORE PARAMETER
09CF	65 03	2390		ADC /TEMP1	IN PARAMETER AREA
09D1	A8	2400		TAY	
09D2	A1 08	2410	STORD1	LDA (PARPNT,X)	
09D4	99 00 0B	2420		STA TIME,Y	
09D7	F6 08	2430		INC /PARPNT,X	
09D9	D0 02	2440		BNE NOCAR2	
09DB	F6 09	2450		INC /PARPNT+1,X	
09DD	A1 08	2460	NOCAR2	LDA (PARPNT,X)	
09DF	99 01 0B	2470		STA TIME+1,Y	
09E2	F6 08	2480	FIXUP	INC /PARPNT,X	
09E4	D0 02	2490		BNE NOCAR3	
09E6	F6 09	2500		INC /PARPNT+1,X	
09E8	A6 03	2510	NOCAR3	LDX /TEMP1	
09EA	4C 0C 09	2520		JMP DECR	
09ED	85 03	2530	NREST	STA /TEMP1	SET SO COMMAND
09EF	A9 00	2540		LDA #0	TAKES ZERO TIME
09F1	99 00 0B	2550		STA TIME,Y	
09F4	99 01 0B	2560		STA TIME+1,Y	
09F7	A5 03	2570		LDA /TEMP1	
09F9	C9 C8	2580		CMP #200	
09FB	B0 05	2590		BCS NSTORE	BRANCH IF NOT A
09FD	E9 BF	2600		SBC #191	STORED COMMAND
09FF	0A	2610		ASL A	
0A00	D0 CB	2620		BNE STORD	
0A02	84 03	2630	NSTORE	STY /TEMP1	
0A04	D0 13	2640		BNE NOCHAN	BRANCH IF NOT "CHANNEL"
0A06		2650	* PROCESS	CHANNEL COMMAND	
0A06	A1 08	2660		LDA (PARPNT,X)	
0A08	29 F3	2670		AND #\$F3	

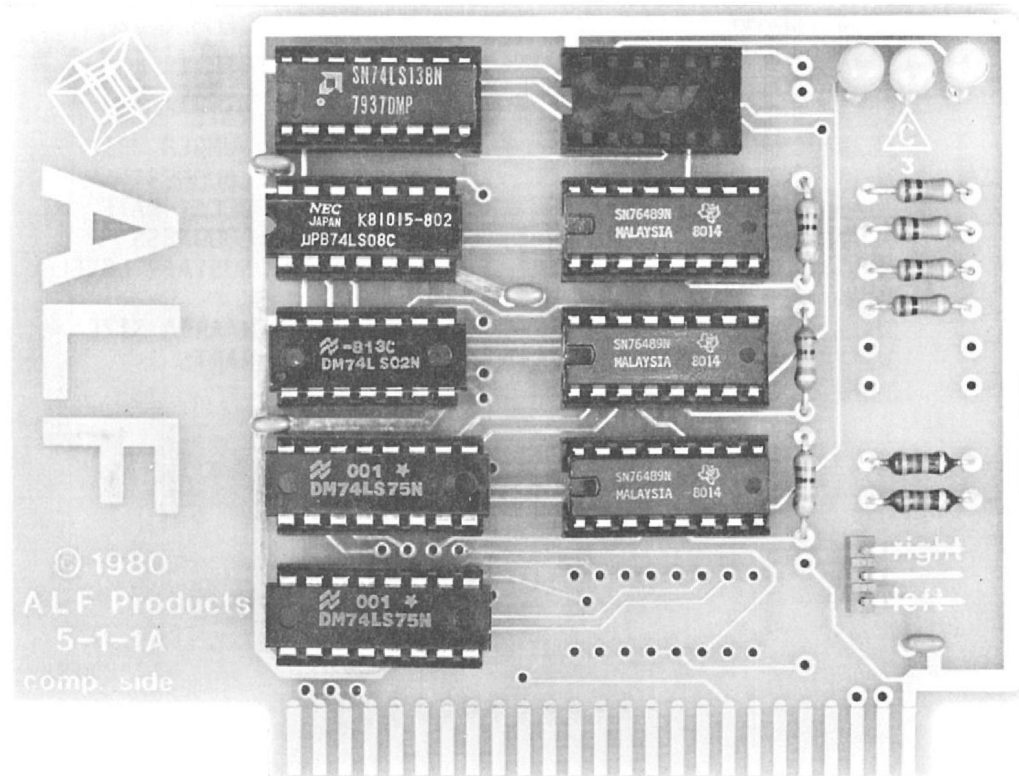
0A0A	99	10	0B	2680	STA CHAN,Y	
0A0D	A1	08		2690	LDA (PARPNT,X)	
0A0F	0A			2700	ASL A	
0A10	0A			2710	ASL A	
0A11	0A			2720	ASL A	
0A12	09	9F		2730	ORA #\$9F	
0A14	99	11	0B	2740	STA CHAN+1,Y	
0A17	D0	76		2750	BNE NOP	
0A19	C9	CA		2760	NOCHAN CMP #202	
0A1B	B0	33		2770	BCS NOCALL	BRANCH IF NOT A "CALL"
0A1D				2780	* PROCESS CALL COMMAND	
0A1D	A1	08		2790	LDA (PARPNT,X)	COMPUTE CALLED ADDRESS
0A1F	65	00		2800	ADC /SPNTR	
0A21	85	05		2810	STA /TEMP3	
0A23	F6	08		2820	INC /PARPNT,X	
0A25	D0	02		2830	BNE NOCAR4	
0A27	F6	09		2840	INC /PARPNT+1,X	
0A29	A1	08		2850	NOCAR4 LDA (PARPNT,X)	
0A2B	65	01		2860	ADC /SPNTR+1	
0A2D	85	06		2870	STA /TEMP3+1	
0A2F	8A			2880	TXA	STORE RETURN ADDRESS
0A30	A8			2890	TAY	
0A31	B5	08		2900	LDA /PARPNT,X	
0A33	69	01		2910	ADC #1	
0A35	91	05		2920	STA (TEMP3),Y	
0A37	C8			2930	INY	
0A38	B5	09		2940	LDA /PARPNT+1,X	
0A3A	69	00		2950	ADC #0	
0A3C	91	05		2960	STA (TEMP3),Y	
0A3E	A5	07		2970	LDA /PARNUM	ADVANCE CALLING
0A40	0A			2980	ASL A	ADDRESS OVER
0A41	65	05		2990	ADC /TEMP3	RETURN ADDRESSES
0A43	95	08		3000	STA /PARPNT,X	
0A45	A5	06		3010	LDA /TEMP3+1	
0A47	69	00		3020	ADC #0	
0A49	95	09		3030	STA /PARPNT+1,X	
0A4B	A6	03		3040	LDX /TEMP1	
0A4D	4C	0C	09	3050	JMP DECR	
0A50	D0	23		3060	NOCALL BNE NORET	BRANCH IF NOT "RETURN"
0A52				3070	* PROCESS RETURN COMMAND	
0A52	18			3080	CLC	
0A53	A1	08		3090	LDA (PARPNT,X)	COMPUTE RETURN ADDRESS
0A55	65	00		3100	ADC /SPNTR	ADDRESS
0A57	85	05		3110	STA /TEMP3	
0A59	F6	08		3120	INC /PARPNT,X	
0A5B	D0	02		3130	BNE NOCAR5	
0A5D	F6	09		3140	INC /PARPNT+1,X	
0A5F	A1	08		3150	NOCAR5 LDA (PARPNT,X)	
0A61	65	01		3160	ADC /SPNTR+1	
0A63	85	06		3170	STA /TEMP3+1	
0A65	8A			3180	TXA	GO TO
0A66	A8			3190	TAY	RETURN ADDRESS
0A67	B1	05		3200	LDA (TEMP3),Y	
0A69	95	08		3210	STA /PARPNT,X	

```





0A6B C8          3220          INY
0A6C B1 05      3230          LDA (TEMP3),Y
0A6E 95 09      3240          STA /PARPNT+1,X
0A70 A6 03      3250          LDX /TEMP1
0A72 4C 0C 09   3260          JMP DECR
0A75 C9 CD      3270 NORET    CMP #205
0A77 D0 12      3280          BNE NOPERC          BRANCH IF NOT "FUZZ"
0A79           3290 * PROCESS FUZZ COMMAND
0A79 A1 08      3300          LDA (PARPNT,X)
0A7B D0 12      3310          BNE NOP
0A7D F6 08      3320          INC /PARPNT,X
0A7F D0 02      3330          BNE NOCAR7
0A81 F6 09      3340          INC /PARPNT+1,X
0A83 A1 08      3350 NOCAR7   LDA (PARPNT,X)
0A85 99 1A 0B   3360          STA PERCUS,Y
0A88 4C E2 09   3370          JMP FIXUP
0A8B C9 FE      3380 NOPERC   CMP #254
0A8D B0 09      3390          BCS END          BRANCH IF NOT A "NOP"
0A8F F6 08      3400 NOP      INC /PARPNT,X
0A91 D0 02      3410          BNE NOCAR6
0A93 F6 09      3420          INC /PARPNT+1,X
0A95 4C E2 09   3430 NOCAR6   JMP FIXUP
0A98 68          3440 END      PLA          "PROCESS" END COMMAND
0A99 AA          3450          TAX          RESTORE X
0A9A 60          3460          RTS          AND RETURN
0A9B           3470 * SUB-OCTAVE DIVISOR TABLE
0A9B 10 3D      3480 TABLE  DEF 15632
0A9D 50 3B      3490          DEF 15184
0A9F A0 39      3500          DEF 14752
0AA1 00 38      3510          DEF 14336
0AA3 70 36      3520          DEF 13936
0AA5 E0 34      3530          DEF 13536
0AA7 60 33      3540          DEF 13152
0AA9 E0 31      3550          DEF 12768
0AAB 80 30      3560          DEF 12416
0AAD 20 2F      3570          DEF 12064
0AAF C0 2D      3580          DEF 11712
0AB1 70 2C      3590          DEF 11376
0AB3 30 2B      3600          DEF 11056
0AB5 F0 29      3610          DEF 10736
0AB7 C0 28      3620          DEF 10432
0AB9 A0 27      3630          DEF 10144
0ABB 80 26      3640          DEF 9856
0ABD 60 25      3650          DEF 9568
0ABF 50 24      3660          DEF 9296
0AC1 50 23      3670          DEF 9040
0AC3 40 22      3680          DEF 8768
0AC5 50 21      3690          DEF 8528
0AC7 60 20      3700          DEF 8288
0AC9 70 1F      3710          DEF 8048
0ACB           3720 * COMMAND PARAMETER AREA
0B00           3730          ORG $B00
0B00           3740 TIME    BSS 2          TIME REMAINING
0B02           3750 GAP     BSS 2          GAP SIZE

```

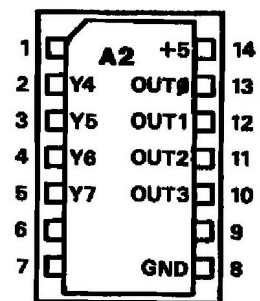
0B04	3760	TRANS	BSS 2	TRANSPOSE VALUE
0B06	3770	ATTACK	BSS 2	ATTACK RATE
0B08	3780	DECAY	BSS 6	DECAY RATE
0B0A	3790	VOLUME	EQU DECAY+2	VOLUME LEVEL
0B0C	3800	SUSTAN	EQU VOLUME+2	SUSTAIN LEVEL
0B0E	3810	RELEAS	BSS 2	RELEASE RATE
0B10	3820	CHAN	BSS 2	CHANNEL NUMBER
0B12	3830	LOUDNS	BSS 2	CURRENT LOUDNESS
0B14	3840	DOWN	BSS 6	CURRENT DECAY RATE
0B16	3850	DESIRE	EQU DOWN+2	DESIRED LOUDNESS
0B18	3860	CURSUS	EQU DESIRE+2	CURRENT SUSTAIN LEVEL
0B1A	3870	PERCUS	BSS 1	FUZZ MASK
0B1B	3880	ASIZE	EQU *-TIME	PARAMETER AREA SIZE
0B1B	3890		BSS ASIZE*8	OTHER 8 PARTS
0BF3	3900		END	

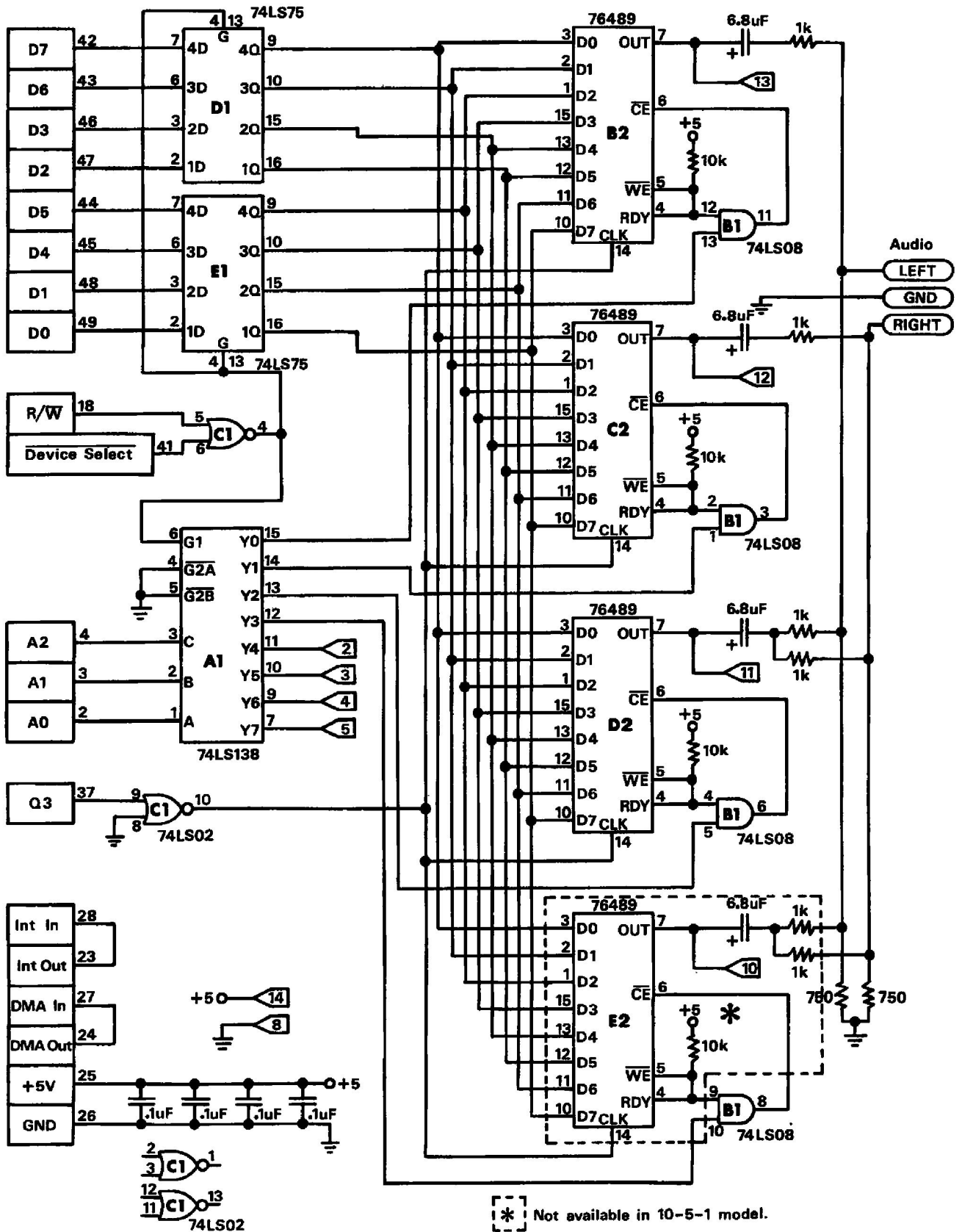


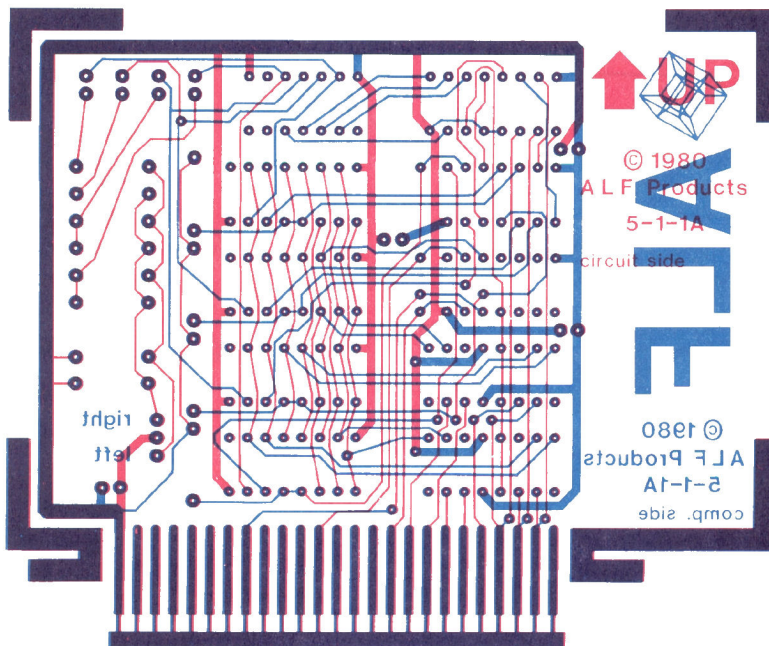
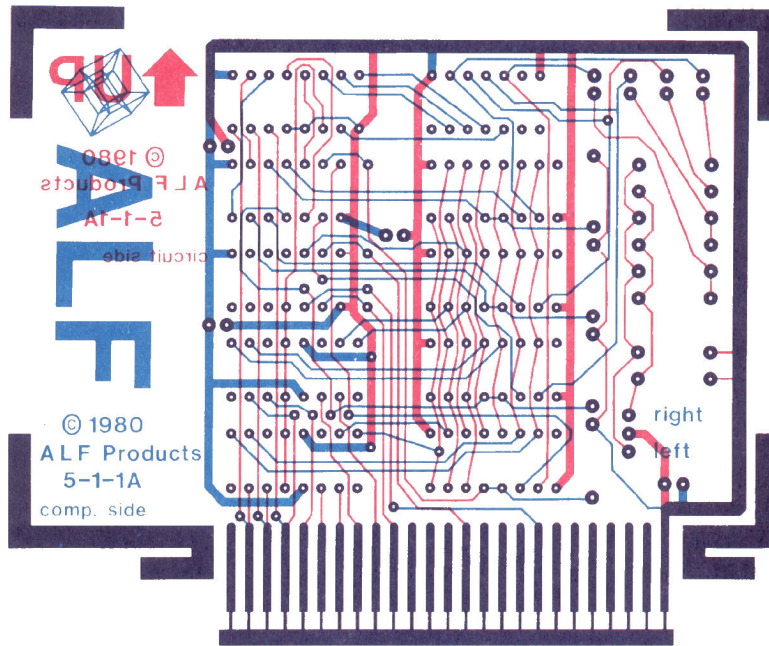
SCHEMATIC TERMINALS

-  Connection to pin on Apple II Peripheral I/O Bus
-  Connection to pin on Access Socket **A2**
-  Connection to Audio Out molex pin
-  Connection to power bus

ACCESS SOCKET







INDEX

- ***DISK: 2-32
- *: 2-3, 2-4
- .: 2-6, 2-8, 2-37
- 1Ø-1-2 Mono Cable: 1-1
- 3: 2-17, 2-37
- 65Ø2 Programming: 6-2
- Accidental Control Symbols: 2-3Ø
- Addresses, Selected Hex: 2-41
- ALBUM File: 3-1, 4-1, 4-2
- "America": 2-1 to 2-12
- Applesoft: 1-1 to 1-2, 2-4Ø, 5-1, 5-4 to 5-5, 5-6, 5-7
- Asterisk: 2-3, 2-4
- ATTACK: 2-22, 2-33, 2-37, 5-11, 5-14, 5-15 to 5-17
- Audio Cable: 1-1, 1-2
- Backup: 1-3, 2-4Ø
- Beaming: 2-27
- Cable, Audio: 1-1, 1-2
- CALL: 2-18, 2-37, 5-9, 5-12, 5-13, 5-14, 5-17
- CHANNEL NUMBER: 5-9, 5-12, 5-14
- Circuit Card Photo: 7-9
- Command Numbers: 5-14
- Commands in ENTRY: 2-29 to 2-4Ø
- Commands, Type 1: 2-29 to 2-32
- Commands, Type 2: 2-29, 2-32 to 2-35
- Commands, Type 3: 2-29, 2-35 to 2-36
- Commands, Type 4: 2-29, 2-36 to 2-39
- Commands, Type A: 5-9 to 5-1Ø
- Commands, Type B: 5-1Ø to 5-11
- Commands, Type C: 5-11 to 5-12
- Configuration, Song: 5-6
- Converting a Song to a Binary File: 5-1 to 5-2, 5-7
- Copying Songs without ENTRY: 2-4Ø
- Correcting Mistakes: 2-12 to 2-15
- Current Decay: 2-24 to 2-27, 5-12, 5-15 to 5-17
- Current Loudness: 2-23 to 2-27, 2-37, 2-38, 2-39, 5-15 to 5-17
- Current Sustain: 2-24 to 2-27, 5-12, 5-15 to 5-17
- Cursor: 2-2 to 2-8, 2-15
- DECAY: 2-22, 2-33, 2-37, 5-11, 5-12, 5-14, 5-15 to 5-17
- DEL: 2-13, 2-3Ø, 2-32
- DELETE: 2-32 to 2-33
- Desired Loudness: 2-23 to 2-27, 2-38, 2-39, 5-12, 5-15 to 5-17
- DISCO: 3-1, section 4
- Disk Software: 1-3
- Divisor Calculation: 6-2 to 6-3
- Dot: 2-6, 2-8, 2-37
- Duration: 2-29 to 2-3Ø
- EDIT: 2-9, 2-15, 2-33, 2-34
- END: 5-1Ø, 5-12, 5-14
- END, in DISCO: 4-2
- END Marker: 2-4, 2-5
- Entering a Simple Song: 2-1 to 2-12
- Entering Rests: 2-15 to 2-16
- ENTRY: section 2, 5-1, 5-11
- ENTRY, operating tips: 2-39
- ENTRY, selected hex addresses: 2-41
- ENTRY, summary of commands: 2-29 to 2-4Ø
- Envelopes: 2-21 to 2-27, 2-37 to 2-38
- Enveology: 2-23
- EXEC Files: 3-1, 4-1
- Fixed Playback Speed: 5-7
- Flat: 2-35, 2-37
- FP: 1-1 to 1-2, 3-1, 3-2, 4-1, 5-2, 5-3
- Free Notes: 2-1, 2-34 to 2-35, 2-36
- Frequency: 6-1, 6-2 to 6-3
- FUZZ: 2-34, 2-38, 5-11, 5-14
- GAP: 2-22, 2-33, 2-38, 5-11, 5-12, 5-14
- GOTO: 2-3Ø to 2-31
- INS: 2-13, 2-3Ø
- Insertion: 2-5, 2-13
- Installation: section 1
- INT: 1-2, 3-1, 3-2, 4-1, 5-2, 5-3
- INTEGER: 2-31
- Integer/Applesoft Switch: 2-4Ø
- Integer BASIC: 1-1 to 1-2, 2-4Ø, 2-41 5-3 to 5-4, 5-6, 5-7 to 5-8
- KEY: 2-2, 2-33, 2-35, 2-41
- Key Signature: 2-7, 2-37
- Left & Right Movement Controls: 2-4, 2-5, 2-3Ø
- LENGTH: 2-31, 2-37
- Line 1Ø: 1-2, 1-3, 2-1, 3-1, 5-8
- LOAD: 2-19, 2-33, 3-1

Loading & Saving Songs: 2-19 to 2-20
Loudness, definition: 2-21

Measure: 2-6, 2-36, 2-37
MEASURE: 2-31
Menu Paddle: see Paddle 0
Movement Controls, Left & Right: 2-4,
2-5, 2-30
Music Notation: 2-27

Natural: 2-5, 2-35, 2-37
NEW: 2-2, 2-15, 2-22, 2-33
Note Duration Symbols: 2-29 to 2-30
Note Paddle: see Paddle 1
Notetrinos: 2-24

Operating Tips: 1-2 to 1-3

Paddle 0: 2-3, 2-8
Paddle 1: 2-2, 2-3, 2-36 to 2-37
Paddle 3: 5-7
Paddle Settings: 2-40
Paramatron, High-Power: 2-24
PART: 2-31
Part Data: 5-12 to 5-13
Part, definition: 2-2
Partial Starting Measure: 2-39
Percussion: 2-38, see also FUZZ
PERFORM: section 5
PERFORM, block diagram: 5-15 to 5-17
PERFORM, listing: 7-1 to 7-8
PERFORM, sample sessions: 5-2, 5-7 to
5-8
PERFORM, temporaries: 5-13
PERFORM, technical: 5-9 to 5-14
PITCH: 5-11 to 5-12, 5-14, 5-17
PLAY, command: 2-8, 2-10, 2-31 to 2-32,
3-1
PLAY: section 3
POKE: 2-38
Problem Checklist: 1-3
Programming Bare-Handed: section 6

QUARTER: 2-3, 2-23, 2-33, 2-35, 2-37,
2-41

Recommended Reading: 2-27
Relative Addresses: 5-12
RELEASE: 2-22, 2-33, 2-38, 5-11, 5-14,
5-15 to 5-17
Release Stage: 2-38
Repair: 1-2 to 1-3
Repair Diagram: 7-11
Repeated Sections: see Subroutines
Reset Button: 2-19, 2-35, 2-40, 3-2
Rest: 2-15 to 2-16, 2-36, 5-12, 5-14,
5-17

Rests at the Ends of Parts: 2-39
RETURN: 5-10, 5-13, 5-14, 5-17
Right Movement: 2-4, 2-30
"Row, Row, Row Your Boat": 2-16 to
2-18

SAVE: 2-19, 2-32
Saving Songs: 2-19 to 2-20
Schematic: 7-9 to 7-10
Sharp: 2-35, 2-37
SLOT: 2-1, 3-1, 5-1, 5-8
Song Breakdown, Sample: 2-28
Song, definition: 2-15 to 2-16
Song, entry of: 2-1 to 2-12
Song Configuration: 5-6
Song Data Format: 2-41, 5-12 to 5-14
Speaker/Arrow Symbol: 2-7, 2-30
SPEED: 2-15, 2-23, 2-34, 2-35
START, in DISCO: 4-2
Stereo: 1-1, 2-33, 2-34, 5-9, 6-1
STEREO: 2-15, 2-33, 2-34, 2-38
STOP: 3-1, 5-10, 5-12, 5-14, 5-16
SUBROUTINE: 2-17, 2-34 to 2-35
Subroutines: 2-16 to 2-19, 2-34 to
2-35, 2-37, 5-9 to 5-10
Subroutine Data: 5-13
Suggested Speed, Reading: 5-7
Summary of ENTRY Commands: 2-29 to
2-40
SUSTAIN: 2-22, 2-33, 2-38, 5-11, 5-12,
5-14
Synthesizer Hardware: section 7
Synthesizer Initialization: 5-6, 6-1

Technical, song data: 5-9
Technical, hardware: section 7
TEMPO: 2-39, 5-13, 5-14, 5-17
Tempo, Adjusting: 2-20 to 2-21
Tempo Control: 5-7
Temporaries, ENTRY: 2-41
Temporaries, PERFORM: 5-13
TIE: 2-14, 2-16, 2-36, 2-37
TIME: 2-33, 2-36, 2-41
Time Periods: 2-40
Time Remaining: 5-17
Tips for Operation: 1-2 to 1-3
Tips on ENTRY: 2-39 to 2-40
TRANPOSE: 2-18, 2-33, 2-39, 2-40,
2-41, 5-10, 5-11, 5-14
Triplets: 2-17, 2-37

VOLUME: 2-22, 2-33, 2-39, 5-11, 5-12,
5-14, 5-17, 6-1

White Noise: 2-38

X-Notes: 2-21

