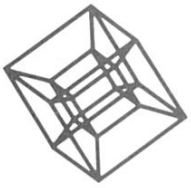
APPLE][® COMPATIBLE

APPLE MUSIC][



A L F PRODUCTS

The

Apple Music][Owner's Manual

Complete Instructions

for the

10-5-1 Apple Music][

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

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

Part Number 11-1-1A

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: 1Ø-5-16 crew

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

CONTENTS

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 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 1Ø. 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 1Ø 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 1Ø you must load the program, change line 1Ø 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 1Ø. 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 1Ø 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:

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.

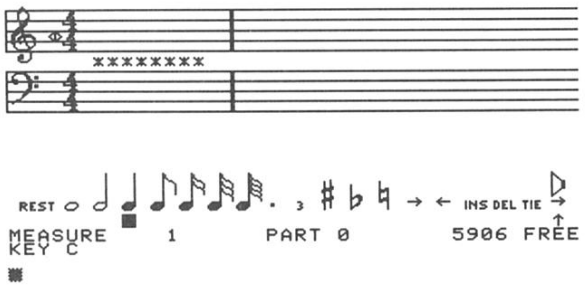
To run this program, you must have 24K or more memory. If you are using a DISK II, you need 32K or more. Very detailed graphics are presented on the screen, so it is recommended that a black and white monitor (such as the Sanyo VM42Ø9 or VM4215) be used rather than a television set, although good results 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 1Ø. It will be 1Ø 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 1Ø. If you ever change the slot position of your synthesizer, you should do this configuration procedure again.

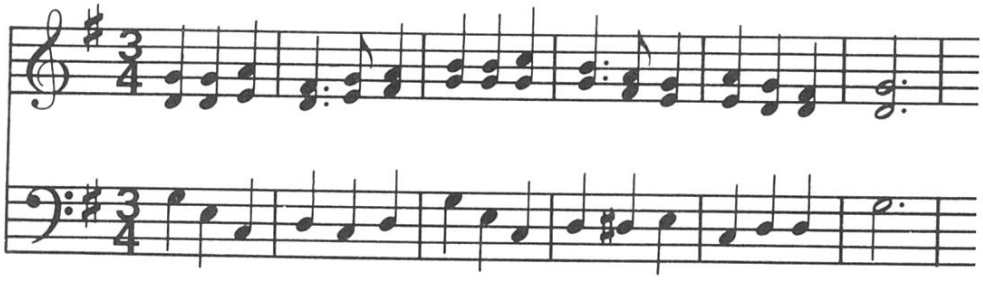
ENTERING A SIMPLE SONG

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:

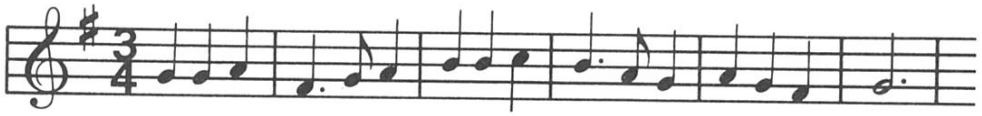


The number in front of "FREE" will vary according to memory size and other factors. It indicates the number of notes which can be added, and will be constantly updated as you enter and edit the song.

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 Ø, is shown below. It is the main melody.

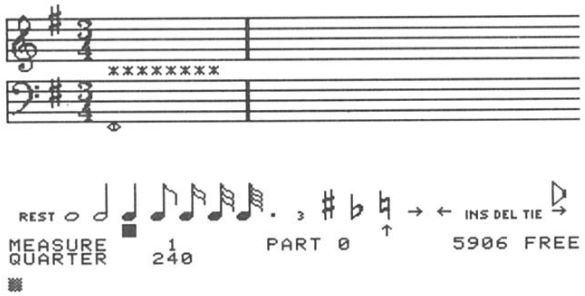


To begin entering a new song, type NEW and press return. ENTRY will display "NUMBER OF PARTS?". Just press return. This will make the song have only 1 part (part Ø). ENTRY now displays "SUGGESTED SPEED?". Since we don't really know what the playback speed should be yet, just press return. ENTRY will 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 Ø 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 Ø 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:

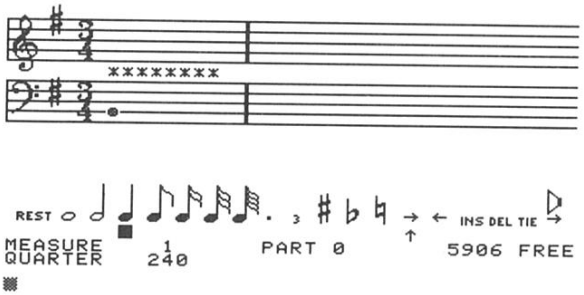


You've only been at this for a few seconds, and already you've told ENTRY two very important facts about "America", the song you're entering. Without these details it would be very difficult to enter the song properly. Why, you're probably half way to being a professional musician, if you weren't one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

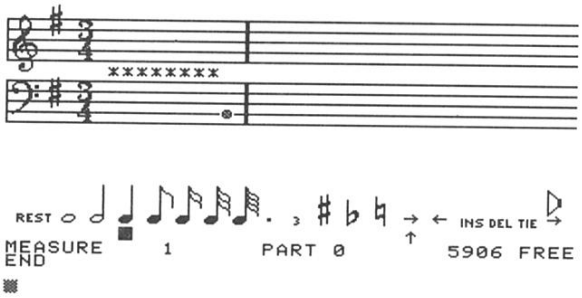
Now the cursor is at the first of eight asterisks (*) displayed between the treble and bass staves, and the item the cursor is at is a QUARTER 24Ø. These eight items are special goodies that describe things about the song which don't display well in sheet music format. This particular one indicates how long a quarter note should play (24Ø 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 Ø 59Ø6 FREE" moves left and right, you're turning paddle Ø, 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 Ø) 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:

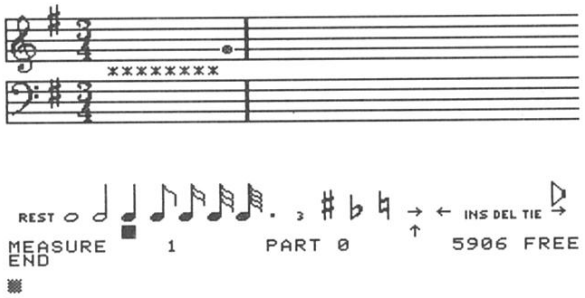


(The position of the flying saucer and the number of notes of FREE space available may be different than shown.) The right pointing arrow is used to move the cursor to the right. To cause a right movement, press the menu button using your left hand. The cursor will move right from an asterisk meaning QUARTER 24Ø to an asterisk meaning GAP 2Ø. To move the cursor right again, press the button with your left hand again. The cursor now moves to TRANSPOSE Ø. Press the button several times. The following items will appear: ATTACK 8192, DECAY 25, VOLUME 55ØØØ, SUSTAIN Ø, and RELEASE 15ØØ. 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 Ø. The screen should now look like this:

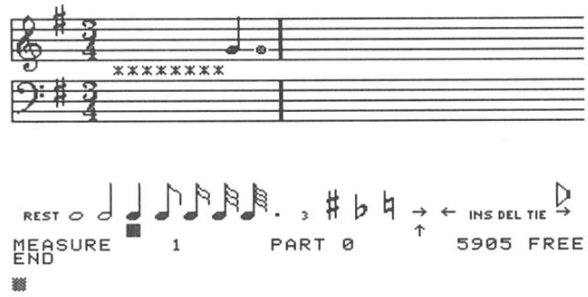


If it doesn't, you probably didn't start with RUN ENTRY like you should have. (The position of the flying saucer and the upward arrow, and the FREE number are not important.) Ready to really get into entering sheet music? Here's part Ø again, just as a reminder:

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 Ø 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:



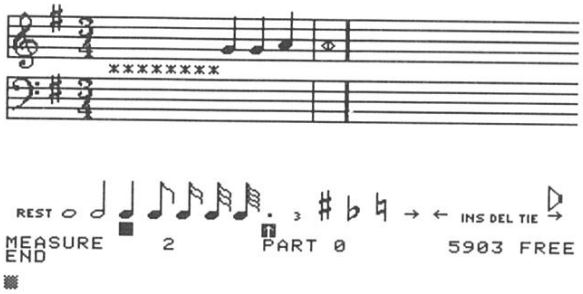
Normally when you type in something like TIME:3/4 or when you press the note button, the time signature (or note or whatever the cursor is pointing at) is written over and thus erased. However, erasing the end marker is not fun, so ENTRY automatically inserts the note (or whatever is entered) in front 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 24Ø 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 24Ø 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 24Ø that said quarter notes should be 24Ø 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.

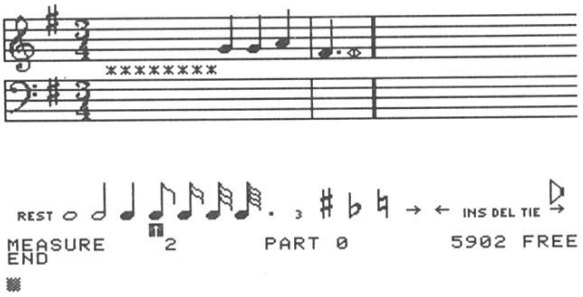
If you're looking ahead at the music for part Ø, then you know that the next note isn't a quarter note. It's a dotted quarter note, which plays for as long as a quarter note plus an eighth note. (A dot always means to add the time of the next shorter note to the note length shown.) You may well be wondering 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:



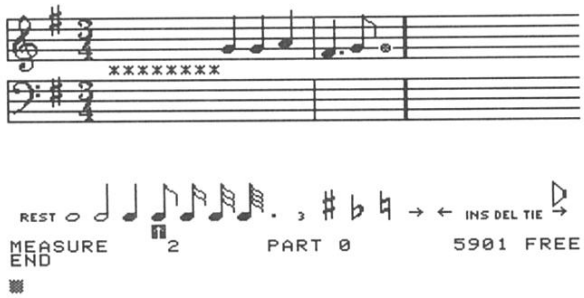
Okay, fire away. Move the note paddle down two clicks to the space under the second treble line, and press the note button. You see how you switch between the left and right hand, usually rotating a knob and pressing a button with the same hand? Since you generally keep your hands on the two knobs, you 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:

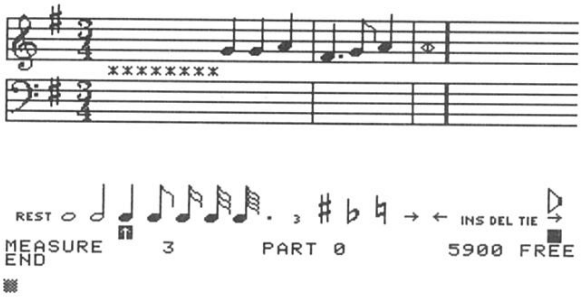


Let's take a look back. Move the cursor left one. (You know how to do it, we just did it a while back to see the first note displayed as NOTE GN3 24Ø.) The eighth note shows up as NOTE GN3 12Ø. It's the same as the first note in this part except it's half as long (only 12Ø time periods). That dotted quarter note we're coming up to should be a quarter (24Ø) plus an eighth (12Ø) long. Back up again to see it. Yep, NOTE FS3 36Ø. 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

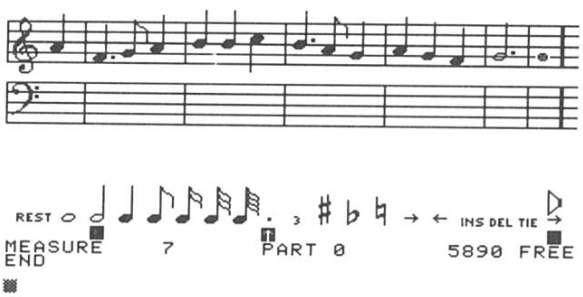
is still dependent on how long you press a button down, but don't worry. It won't be during actual playback. You don't believe me, do you? All right, type PLAY and press return. ENTRY shows "SET SPEED (255) AND PRESS BUTTON". Crank the menu paddle up all the way (it may not actually get up to 255, but 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:



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 Ø 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.

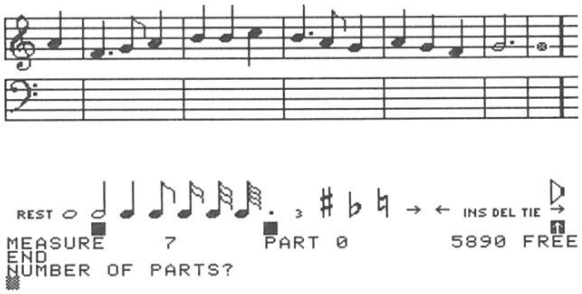
Faster now. Set for dotted quarter. Down a click and punch. Switch to eighth. Down a click and punch. Now quarter. Down a click, punch, up a click, punch, down, punch, down, punch. Last measure. Set for dotted half. (In case you haven't noticed, you can't set for "dot" and then "half" because "half" 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:



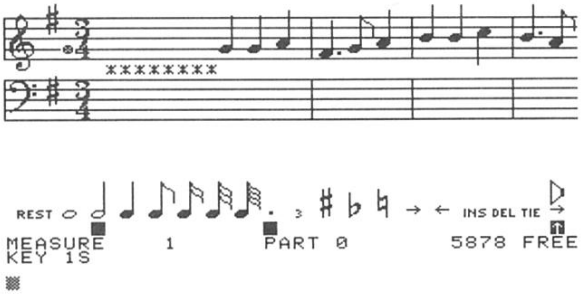
Type PLAY and press return. Let's try a speed of about 2ØØ now. Adjust the menu paddle to some number in the vicinity of 2ØØ. (Don't get too picky, it's not important to get exactly 2ØØ.) Punch the button, and the first 6 glorious measures issue forth.

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!

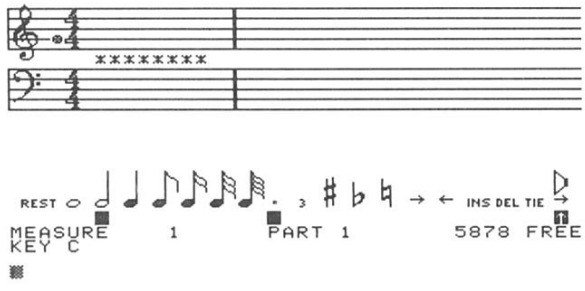
Fortunately, it is quick. First, we have to tell ENTRY that we want to add a second part. Type EDIT and press return. ENTRY responds by showing:



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:



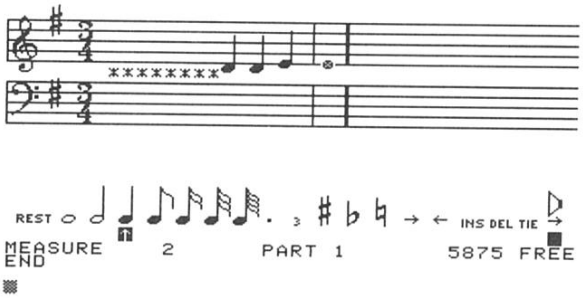
This is the beginning of Part Ø, the part you just entered. The part just created is Part 1. To see Part 1, type PART:1 and press return. The screen shows:



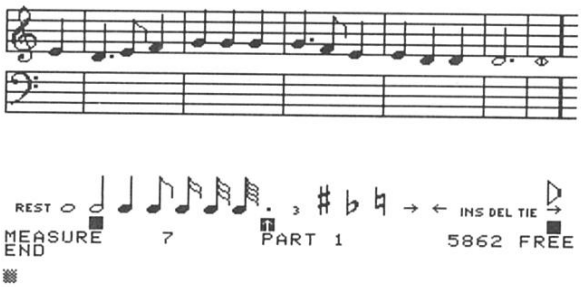
This is just like Part Ø looked originally, except there are fewer notes of

"free" memory, and the screen shows "PART 1" instead of "PART Ø". 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:

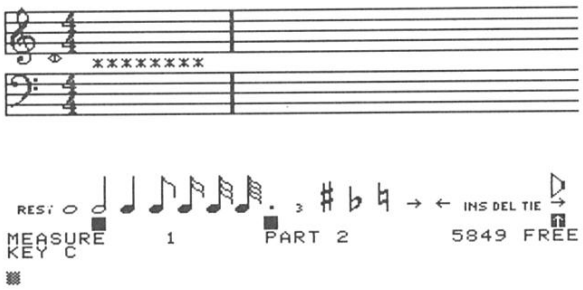


Type PLAY and press return. (As usual, set the speed and press the paddle button to start playback.) You'll notice that only the first measure is played. Playback always stops when the end of the highest numbered part is reached. Since we've only entered the first measure in Part 1, and Part 1 is the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

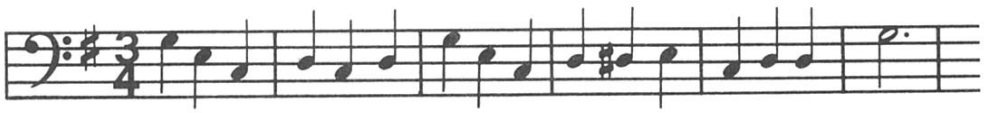


Type PLAY and press return. (I won't tell you to adjust the playback speed paddle since you've got that figured out already.) If there are any wrong notes, back up and correct them. (More details on correcting wrong notes will be given later in this section.) You're now ready to enter the third part.

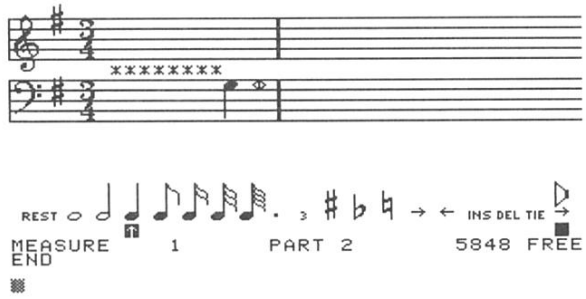
Type EDIT and press return. Ask for 3 parts this time, and then press return to skip the other questions. When Part Ø appears, type PART:2 to go to the third part. The screen shows:



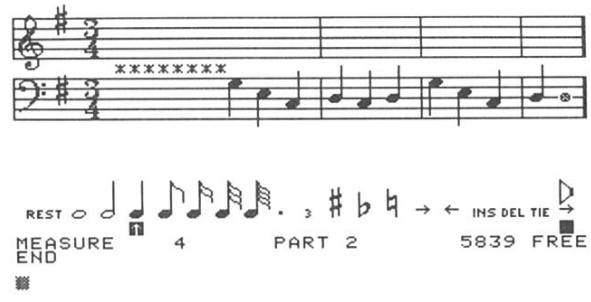
Begin as usual, typing KEY:1S and TIME:3/4, then skip the asterisks. Just for fun, type PLAY and press return. When you press the paddle button to begin playback, there is a brief flash and the hi-res graphics screen reappears. This is because the end of the highest numbered part (now Part 2) is reached immediately, since there are no notes entered in it yet. Now comes your big chance to use the "bass staff", which has been ignored up to this point. The bass staff is the lower five horizontal lines. The sheet music for Part 2 is shown below.



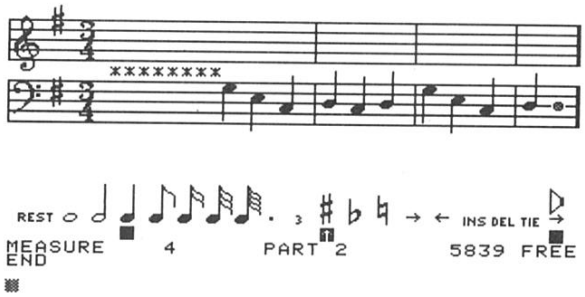
Enter the first note. The screen now shows:



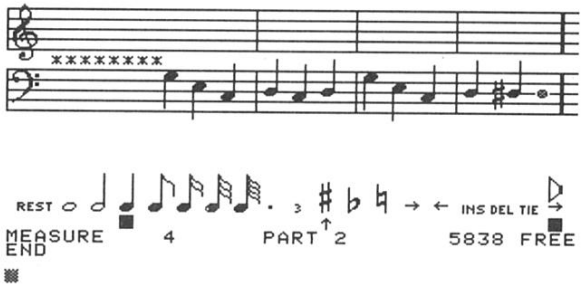
Enter the next nine notes. The screen shows:



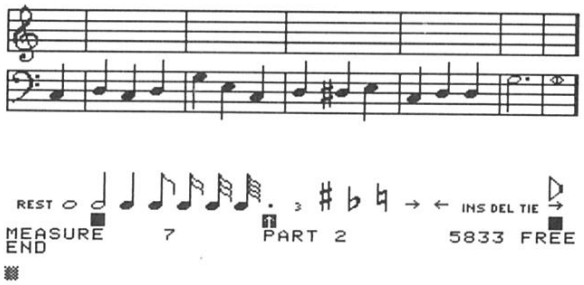
The next note is sharp, so use the menu paddle to light up the sharp sign in the menu, like this:



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



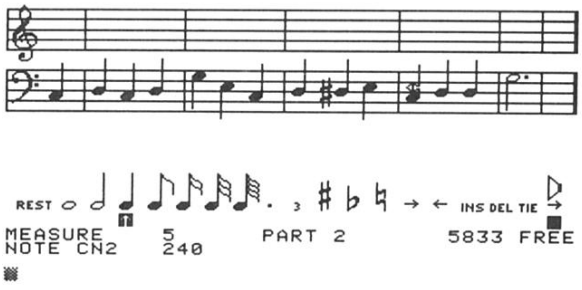
Enter the rest of the part. The screen shows:



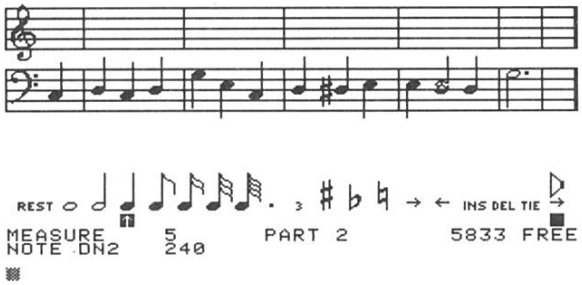
Type PLAY to hear the song and check for errors.

CORRECTING MISTAKES

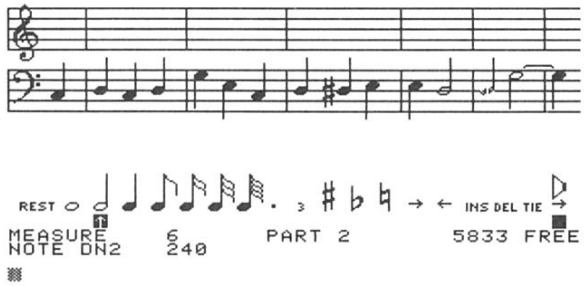
Back up to the first note in measure 5 (of Part 2). Let's say we want to change this note so it is at the next space up on the staff. First, set the menu notes for a quarter note, and put the cursor in the space above the note:



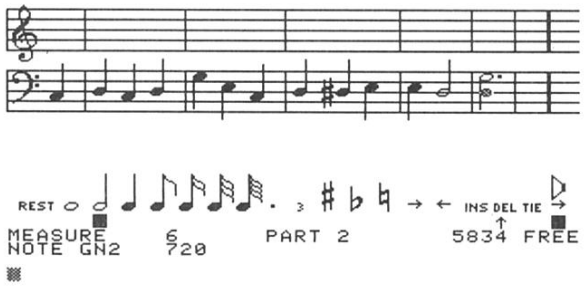
Now just press the note entry paddle button (paddle 1, of course). The old note is written over by the new note:



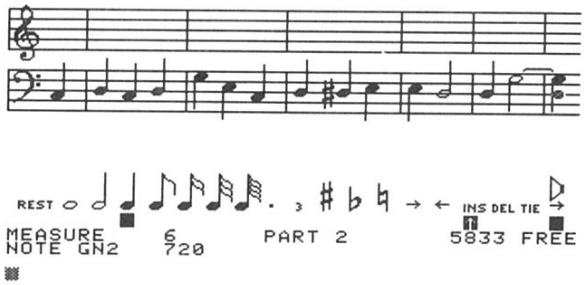
The rest of the song is not affected. Now, let's say we want to change the next note in the measure into a half note of the same pitch. Set for half note, position the cursor so it is over the quarter note's head (in order to get the same pitch), and press the button:



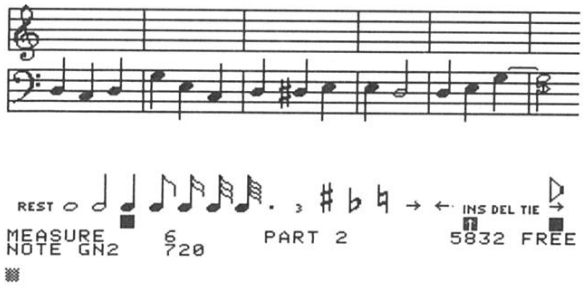
What if we want to get rid of the first note in measure 6 (where the cursor is now)? Just position the arrow for "DEL" and press the menu paddle button:



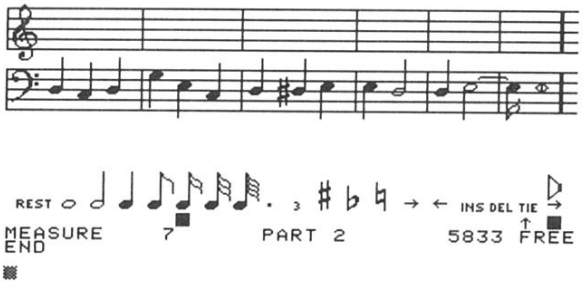
Now, let's change our mind and put it back. It was a quarter note, so set for quarter. Position the cursor on the middle bass staff line to get the same pitch. We need to insert the note, so put the menu arrow under "INS" and press the menu button to light up a block under it. Now just press the note button to enter a note as usual. Instead of replacing the note the cursor is at, the entered note will be inserted in front of it because "insert" mode is on:



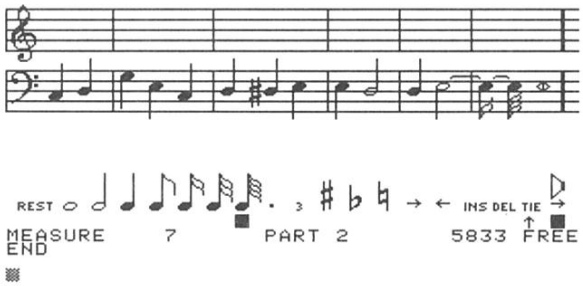
Click the note paddle up one, and press the note button again. Another note is thus inserted:



Now press the menu button while the arrow is pointing at "INS". The block of light goes off. Enter a note. Since "insert" mode is no longer on, the old note is replaced by the new one. Next, back up one and delete the last one of the two similar quarter notes so the next demonstration will be more clear. Let's change the remaining quarter note to a half note. We could set for half note and reenter a half note over the old quarter note, or. . . leave the menu setting at quarter note, aim the menu arrow at "TIE", and press the menu button. There is a beep, and the cursor backs up. Now press the menu button once more to do "TIE" again. The current setting (quarter note) is added to the note the cursor is at. Since it was originally a quarter note and we added a quarter note, it becomes a half note. (Note: the first time you pressed the button for "TIE", the cursor was not at a note or a rest, so the tie could not be done. Since you usually tie the last entered note, ENTRY backs up one when you do an illegal tie, allowing you to just press the button twice to tie the last note.) Now set the menu for a sixteenth note. Aim at "TIE" and press the button twice. The note is now a half note tied to a sixteenth:



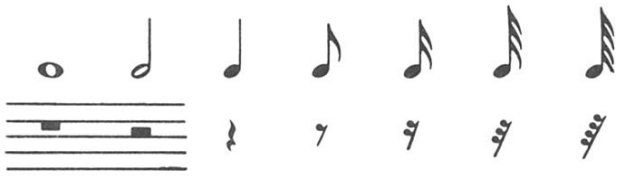
The vertical position of the note paddle cursor is not important during a "tie" since the note paddle is not used. It is important to note that although the half note tied to a sixteenth note is shown as "two" notes, it is really only one. If you back up and look at it, you will see that the length shown 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.

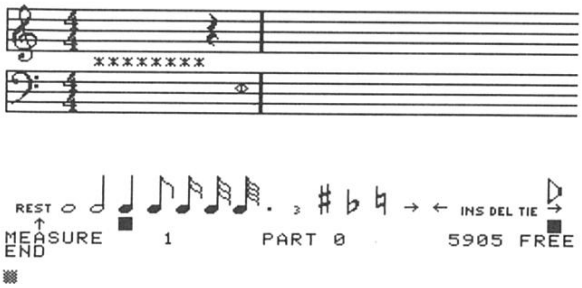
ENTERING RESTS

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:

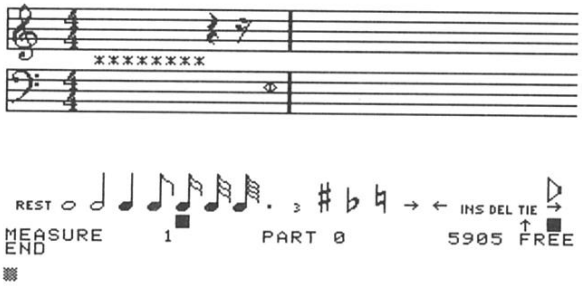


Let's start on a new song. (Actually, "song" refers to a musical composition with lyrics. Technically, one shouldn't use "song" to refer to just any melody, but

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.



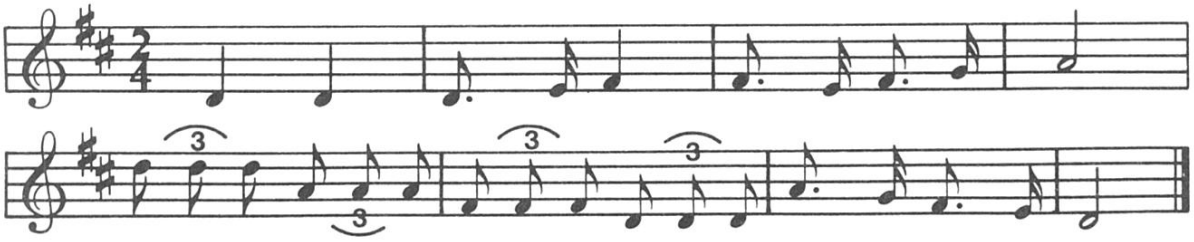
Now select sixteenth note duration and tie it onto the quarter rest. Oddly enough, the screen shows:



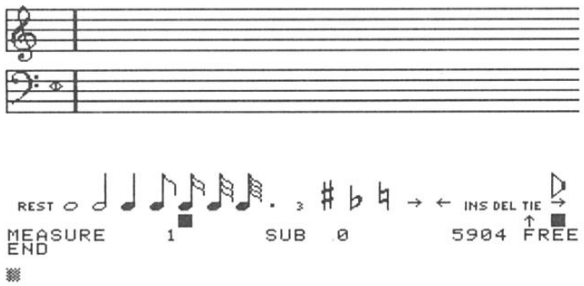
In traditional music notation, rests are never shown as being tied. This is because there is no difference between, for example, a half rest and two quarter rests during performance. The ENTRY screen display makes no distinction between a rest which is as long as a quarter plus a sixteenth, and two rests 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

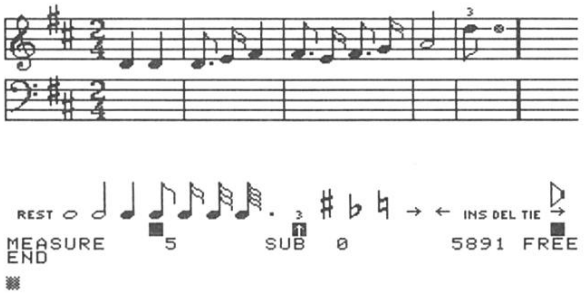
Most people are familiar with the song "Row, Row, Row your Boat". If you're not, become so. This song plays the same theme several times, and from several parts. It seems that one would have to enter this theme several times. Since repeated sections such as this are common in music, ENTRY has special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



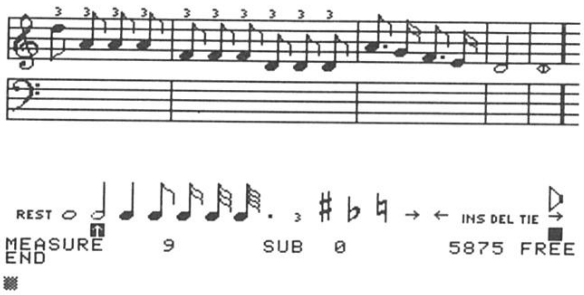
This theme must be entered in a special fashion which allows it to be played many times. This is done using a subroutine. Type NEW and press return several times (as usual) to start fresh. Now type SUBROUTINE:Ø and press return. The screen will show:



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 TRANSPOSE Ø is shown. If we played the basic theme exactly the same in both parts, they would be hard to tell apart. So, type TRANSPOSE:24 and press return. The TRANSPOSE Ø is of course thus changed to TRANSPOSE 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 TRANSPOSE: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 TRANSPOSE 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 TRANSPOSE: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 1ØØ subroutines numbered Ø through 99. Always begin with subroutine Ø 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 Ø 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.

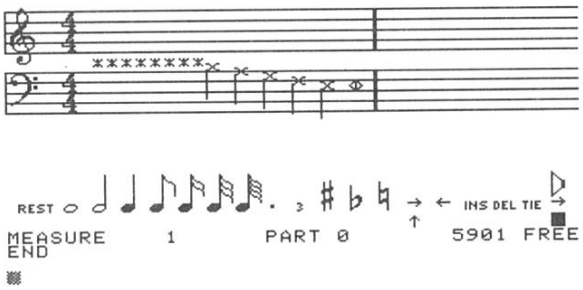
The synthesizer is supplied with a few sample songs which can be loaded and played. Additional songs are available at extra cost.

ADJUSTING THE TEMPO

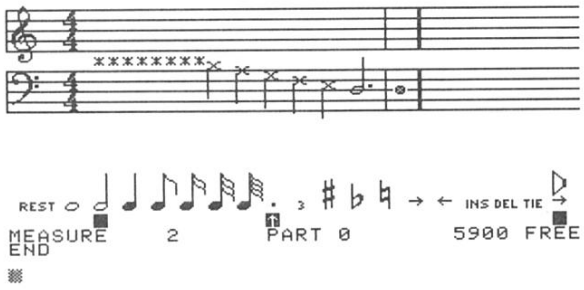
Let's say we want to enter the "row" theme to play twice as fast with the same paddle setting. That means each note will have to play for half as many time periods. Type NEW and press return as required, enter the key and time signatures, and you'll be at the QUARTER 24Ø function. Type QUARTER:12Ø. This will make all quarter notes be entered as 12Ø time periods instead of 24Ø (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:Ø to get back to the beginning of the part, and skip over to the QUARTER function. Change it back to QUARTER:24Ø. 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:12Ø. Notes before the QUARTER function will be shown as half as long as originally entered due to the QUARTER:24Ø, and notes after the QUARTER:12Ø 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:12Ø as it was when the notes were orignially 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:24Ø 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 occured, 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:78ØØ. 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:24Ø settings, a whole note (96Ø time periods) played with a GAP setting of 24Ø would have three quarter notes (96Ø-24Ø, or 72Ø time periods) worth of attack, decay, and sustain; then one quarter note (24Ø 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 Ø to 65535. This number divided by 4Ø96 is the actual volume setting on the synthesizer at the moment. The desired loudness is also a number from Ø 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.

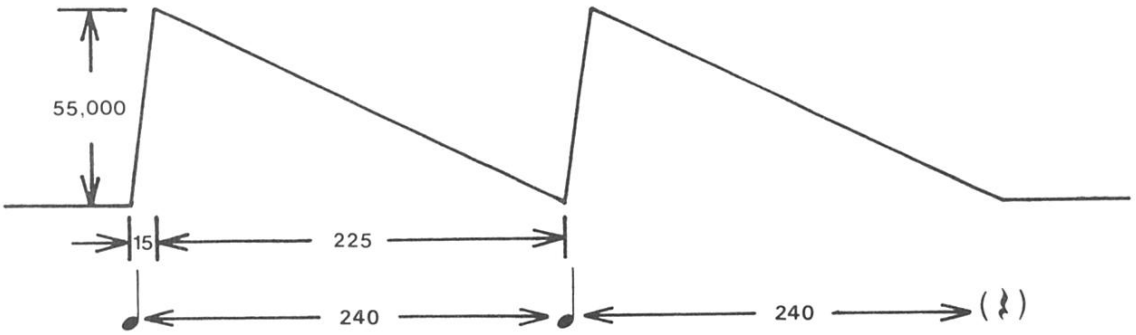
Notetrinos generated using a high-power paramatron at the University of Northern South Dakota (just across the border from Hoople) have revealed the following characteristics of these settings. (What?) When a new note begins, the most recent decay setting is written into the "current decay" rate, the 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.

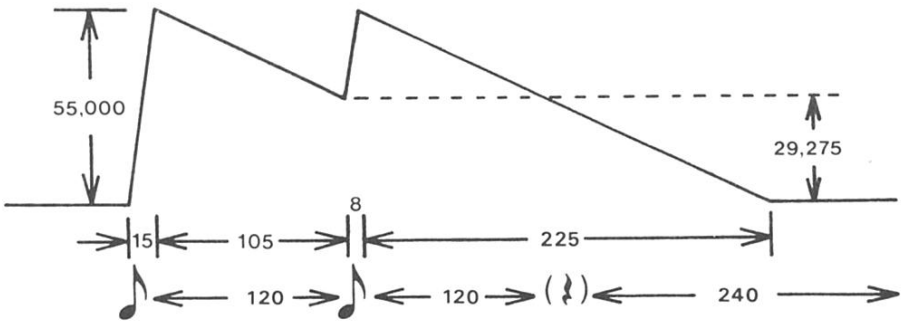
This simple process generates a variety of complex envelopes, for single notes or for several. Be ye not confused: each note does not necessarily have an "attack" and "decay" stage (and so forth). In fact, if the current loudness is

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 55ØØØ. Further, let's say it is a quarter note with standard QUARTER settings (24Ø 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 Ø to 55ØØØ in 24Ø/16 time periods. 55ØØØ/(24Ø/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 55ØØØ back down to Ø in 24Ø*15/16 time periods. 55ØØØ/(24Ø*15/16) is 244.44 so we want a decay setting of 245. The loudness contour will appear thus:

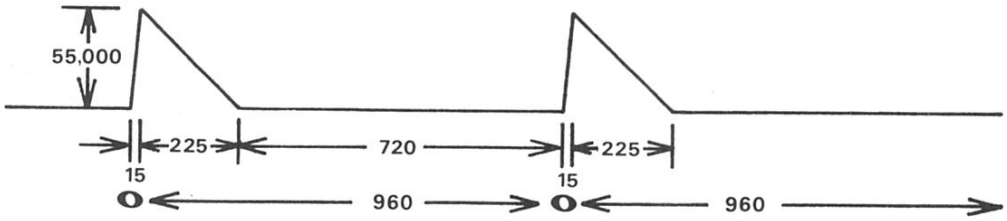


The GAP setting must be 65535 to avoid a release stage. Now, what if we played an eighth note with this setting? The loudness contour would appear thus:

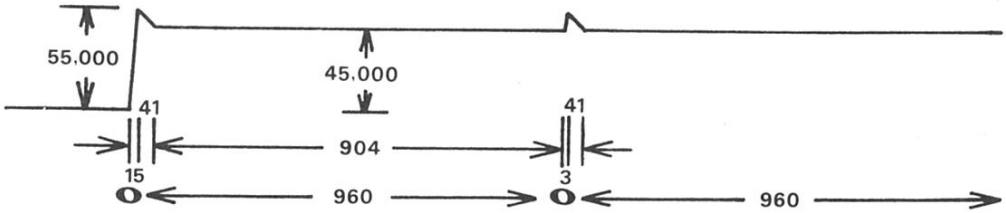


If an eighth note is followed by a rest, the release stage will begin. Therefore

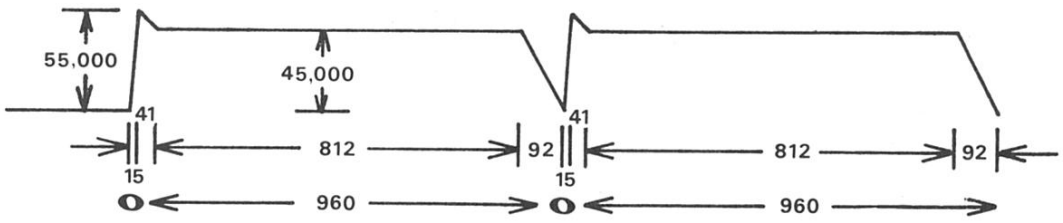
the release setting should be set to the same as the decay setting, unless you want something different to happen on notes followed by rests. What if we played a whole note? Behold:



This assumes the sustain level was set to Ø. What if it were 45ØØØ?:



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:49Ø. Now, it will take 45ØØØ/49Ø time periods for the current loudness to drop from 45ØØØ (the sustain level) to Ø, so we need a GAP setting of 45ØØØ/49Ø (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

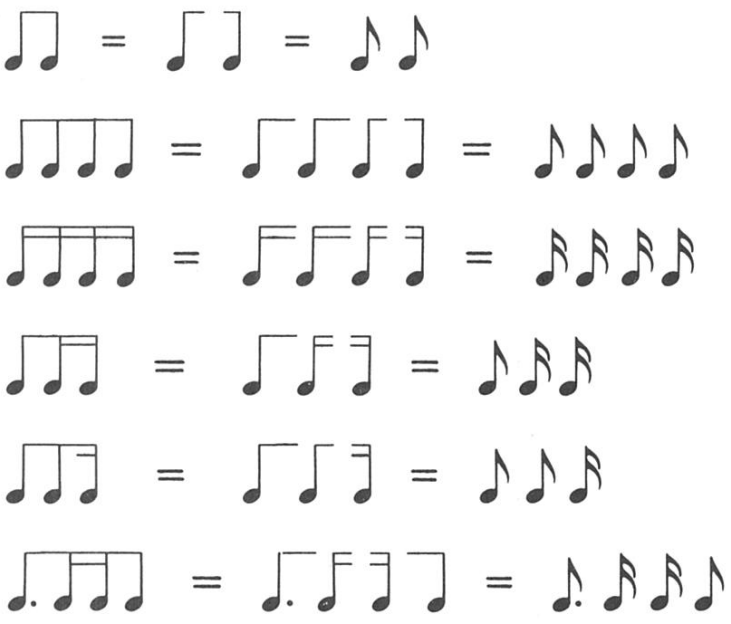
short rest before the CALL in one of the parts to delay it slightly (for an "echo" effect) or both.

RECOMMENDED READING

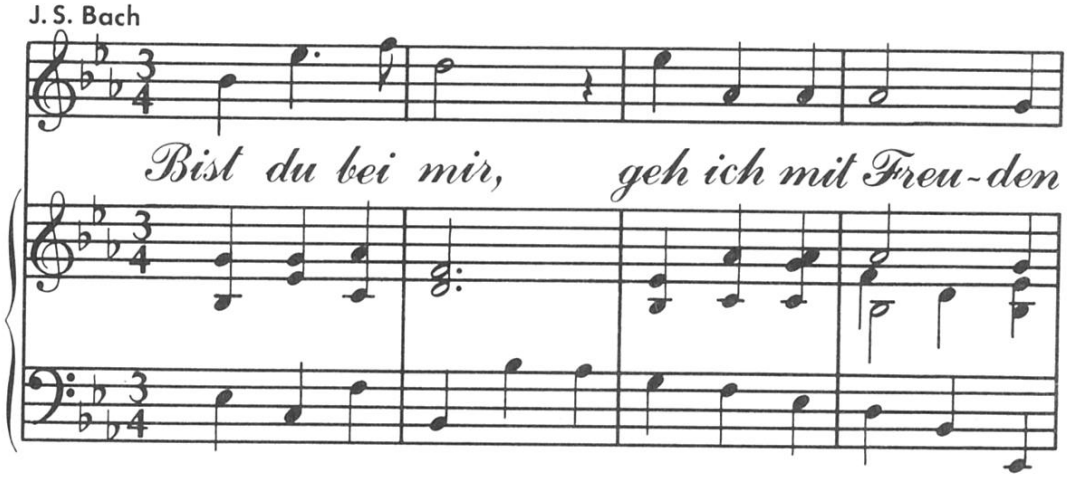
For those of you who are unfamiliar with standard sheet music notation, or for those who encounter some particularly obscure notation, there is an excellent book which you can order from any bookstore. Just ask your local store to order "Music Notation, A Manual of Modern Practice" by Gardner Read, Taplinger Publishing Co. ISBN Ø-8ØØ8-5453-5. In the unlikely event that you have no local bookstores, you can order it from ALF (part number 11-2-1).

BEAMING

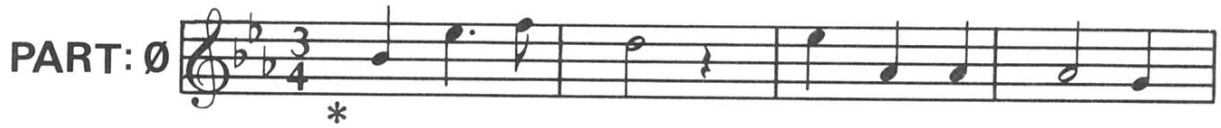
IF WRITTEN: **ENTER:**



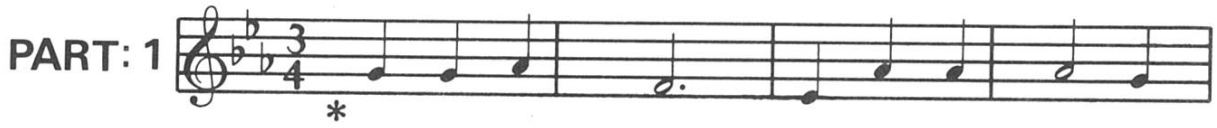
etc.

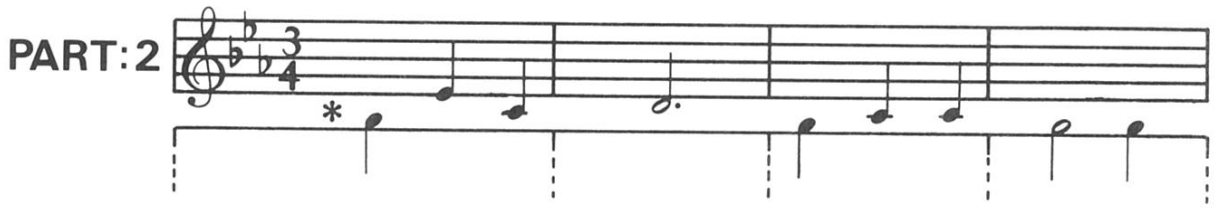


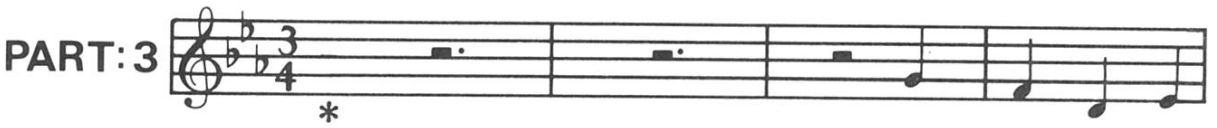
BECOMES:

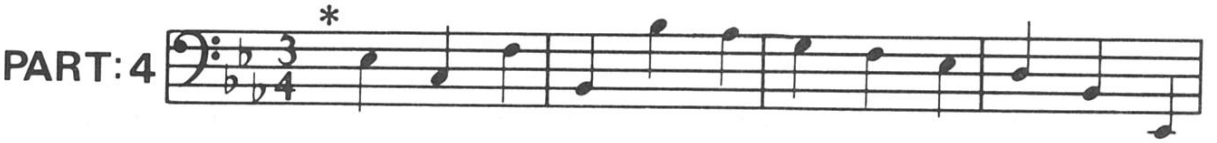


(additional *'s ommited for clarity)









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:55ØØØ. 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:Ø instead of SUBROUTINE:Ø. 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 Ø'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 Ø'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 Ø'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 Ø'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 Ø'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:<Ø-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, 36864-4Ø959 orange, 4Ø96Ø-45Ø55 grey, 45Ø56-49151 pink, 49152-53247 green, 53248-57343 yellow, 57344-61439 aqua, 6144Ø-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 Ø 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 1Ø-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 Ø'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 4Øth character, are set to space. The SUGGESTED SPEED must be from Ø to 255. (1 through 255 select paddle speeds to be suggested before song playback. Ø 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 1Ø. 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 Ø. 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 Ø. 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:24Ø, GAP:2Ø, TRANSPOSE:Ø, ATTACK:8192, DECAY:25, VOLUME:55ØØØ, SUSTAIN:Ø, and RELEASE:15ØØ. (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 Ø 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 Ø. 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 24Ø divided by 5Ø (using SPEED:1/5Ø) would become 4 since 24Ø/5Ø 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 Ø 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 Ø, 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 Ø 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:<Ø-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 Ø up (by ones) when a song is loaded and when RESET is pressed (CØØG 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 occured 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 occured in the part or subroutine so far, it is assumed to be QUARTER:24Ø. All subroutines should start with a QUARTER command if the SPEED command is to be

used.) See the PADDLE Ø and PADDLE 1 Type 4 Commands for additional details. Sample command: QUARTER:48Ø (return).

TIME:<1-19>/<note>
The TIME command is used to change the time signature. (If no TIME command has occured 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 Ø to 65535, optionally followed by a slash (/) and another integer from Ø 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 Ø'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

TRANSPOSE (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 Ø'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:55ØØØ/3Ø (return).

CALL:<Ø-99>

The CALL command is used to have the Type 4 Commands in the specified subroutine be executed during playback. The integer (from Ø 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:1ØØ (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:6Ø (return).

POKE:<Ø-255>,<Ø-255>,<Ø-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 3Ø to 175 followed by Ø and Ø (for example, POKE:78,Ø,Ø) 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:144,24Ø,Ø (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:1ØØ (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 Ø for a release stage or the current volume setting for an attack stage. Sample command: SUSTAIN:45ØØØ (return).

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

TRANSPOSE:<Ø-255>
The TRANSPOSE command is used to change the current transpose setting. Values from Ø to 127 raise all following pitches (until the next TRANSPOSE command) by Ø 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: TRANSPOSE: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:5ØØØØ (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 15Ø 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 93ØØØ/(<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 accidently delete half the melody.

TRANSPOSE

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

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, CØØG (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 5ØØØ hex.
2. The END command (FF ØØ ØØ) is followed by a byte giving the suggested speed, then 16Ø 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 Ø 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 TRANSPOSE 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)
5ØØØ: 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
4DCØ-4DCB: standard stereo positions
Base page usage: (see also PERFORM base page usage)

Ø-19 26-27 36-39 3C-3F 4A-4D 5Ø-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 1Ø. It will be 1Ø 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 1Ø. 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 3DØG (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 1Ø for this program.) Type RUN 1ØØØ 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 <u>not</u> type RUN PLAY), otherwise type INT or FP. If you wish to hear the songs again, type EXEC ALBUM (or RUN DISCO) <u>after</u> typing INT or FP.

TO ADD SONGS

Load the DISCO program, and type RUN 2ØØØ (return). After the instructions are printed, proceed in the same fashion as when originally creating the album (done with RUN 1ØØØ, 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 1ØØØ 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 1ØØØ or RUN 2ØØØ), 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 2ØØØ) 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 1ØØØ. 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 1ØØØ or RUN 2ØØØ) 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 2Ø5Ø decimal (8Ø2 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 2Ø48 decimal (8ØØ hex) you must put the starting address of the song MOD 256. (In Applesoft, this is address-INT(address/256)*256 since MOD is not available.) At 2Ø49 decimal (8Ø1 hex) you must put the starting address of the song divided by 256. (In Applesoft, this is INT(address/256).) Then, a CALL to 2Ø5Ø decimal (8Ø2 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,A2Ø5Ø,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,A2Ø5Ø to load it, and then BSAVE PERFORM,A2Ø5Ø,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.

```
1Ø POKE 76,Ø : POKE 77,124 : DIM A$(4Ø) : INPUT "SONG NAME?",A$
2Ø PRINT "dLOADM:";A$ : A=PEEK(2Ø2)+PEEK(2Ø3)*256
3Ø PRINT "dBSAVE";A$;",A";A;",L";31744-A : PRINT "LENGTH: ";31744-A
4Ø PRINT "dINT"
```

Note: this program requires 48K. On 32K systems, change the 124 to an 8Ø and the two 31744's to 2Ø48Ø'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.

```
1Ø POKE 76,PEEK(115) : POKE 77,PEEK(116) : POKE 217,Ø
2Ø HIMEM:3ØØØ : INPUT "SONG NAME?";A$
3Ø POKE PEEK(54)+PEEK(55)*256+3Ø65,Ø
4Ø PRINT "dLOADM:";A$ : A=PEEK(2Ø2)+PEEK(2Ø3)*256
5Ø L=PEEK(76)+PEEK(77)*256-A : PRINT "dBSAVE";A$;",A";A;",L";L
6Ø 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,A3Ø59 and then POKE 2Ø48,243 and POKE 2Ø49,11. Type CALL 2Ø5Ø to play the song. Note that paddle Ø controls the playback speed. When playback is finished, you could play the song again just by typing CALL 2Ø5Ø.

What are the mystic pokes for? Locations 2Ø48 and 2Ø49 must be set to the starting memory address of the song data. We loaded the song at 3Ø59. Note that 11*256+243 (11 and 243 being the numbers we poked) is 3Ø59, the starting address. 3Ø59 just happens to be the first byte of memory available after PERFORM, which uses locations 2Ø48 through 3Ø58.

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 3Ø59 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 (3Ø59). 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 MOD 256 and LOMEM/256 are (for the new LOMEM, of course). For example, if your longest song is less than 2Ø48 bytes, LOMEM could

be 3Ø59+2Ø48 which is 51Ø7. 51Ø7 MOD 256 is 243 and 51Ø7/256 is 19. To have your program change LOMEM and CM to these values, make the first statements POKE 74,243 : POKE 2Ø4,243 : POKE 75,19 : POKE 2Ø5,19. 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,A3Ø59" where "song name" is the name of the song to be loaded. If you wish to load a second song, change the 3Ø59 after the A to a value which is 3Ø59 plus the length of the first song or greater. Similarly, a third song can be loaded with an A value of 3Ø59 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 2Ø48, and the address/256 must be poked at 2Ø49. Then a CALL 2Ø5Ø 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 2Ø5Ø 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 3Ø59 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 4Ø 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 1Ø3,182 : POKE 1Ø4,12. 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(2Ø5Ø)<>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 2Ø48 and 2Ø49. The starting address is the value A (or A+L) in the BLOAD command. Use POKE 2Ø48,A-INT(A/256)*256 : POKE 2Ø49,A/256. Then a CALL 2Ø5Ø 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 2Ø5Ø 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 1Ø3,182 : POKE 1Ø4,12 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 2Ø48,A MOD 256 : POKE 2Ø49,A/256 : CALL 2Ø5Ø : RETURN (which is POKE 2Ø48,A-INT(A/256)*256 : POKE 2Ø49,A/256 : CALL 2Ø5Ø : 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 "dBLOAD 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(-2192Ø)+PEEK(-21919)*256+A-161). Note that the entire song must be located below memory address 32768 when using Integer BASIC. The -2192Ø is for a 48K system and must be -383Ø4 on a 32K system. Likewise, the -21919 must be -383Ø3 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 Ø to control the playback speed, you must POKE 2346,n where n is the paddle number plus 1ØØ. (For a fixed playback speed, you may wish to install a 15ØK 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.

>BSAVE PERFORM,A2Ø5Ø,L713
>INT
```

CONVERT PERFORM TO A BINARY FILE

```
>1Ø POKE 76,Ø : POKE 77,124 : DIM A$(4Ø) : INPUT "SONG NAME?",A$
>2Ø PRINT "dLOADM:";A$ : A=PEEK(2Ø2)+PEEK(2Ø3)*256
>3Ø PRINT "dBSAVE";A$;",A";A;",L";31744-A : PRINT "LENGTH: ";31744-A
>4Ø PRINT "dINT"
>SAVE CONVERT
```

SAVE CONVERT PROGRAM

```
>RUN
SONG NAME?DIXIE BOOGIE

LENGTH: 3298

>INT
```

CONVERT DIXIE BOOGIE TO BINARY

```
>6 POKE 74,Ø : POKE 2Ø4,Ø : POKE 75,64 : POKE 2Ø5,64   PROTECT PERFORM
>1Ø SLOT=4   (CHANGE AS REQUIRED)                       AND SONG AREA
>3Ø PRINT "dBLOAD PERFORM" : DIM A$(4Ø)    LOAD PERFORM
>4Ø PN=SLOT*16-16256      INITIALIZE SYNTHESIZER
>5Ø FOR P=PN TO PN+2
>6Ø POKE P,159 : POKE P,191 : POKE P,223 : POKE P,255 : POKE P,231
>7Ø NEXT P
>8Ø INPUT "SONG NAME?",A$ : A=3Ø59 : PRINT "dBLOAD";A$;",A";A   READ SONG
>9Ø S=PEEK(PEEK(-2192Ø)+PEEK(-21919)*256+A-161)   READ SUGGESTED SPEED
>1ØØ IF NOT S THEN 18Ø
>16Ø PRINT "SUGGESTED SPEED: ";S
>17Ø PRINT PDL(Ø);"  "; : TAB 1 : IF PEEK(-16287)<128 THEN 17Ø
>18Ø FOR B=1 TO PEEK(A)          CONFIGURE SONG
>19Ø PNTR=PEEK(B+B+A-1)+PEEK(B+B+A)*256+A
>2ØØ CHAN=PEEK(PNTR+2) MOD 4 : NAHC=PEEK(PNTR+2) MOD 16 / 4
>21Ø IF NAHC THEN NAHC=3-NAHC
>215 POKE PNTR+1,SLOT*16+CHAN*4+NAHC
>22Ø NEXT B : POKE 2Ø48,A MOD 256 : POKE 2Ø49,A/256
>23Ø CALL 2Ø5Ø : GOTO 8Ø          PLAY THE SONG
>SAVE YALP
>RUN
SONG NAME?DIXIE BOOGIE

SUGGESTED SPEED: 16Ø
(etc.)      (SONG PLAYS WHEN BUTTON IS PRESSED)
```

CREATION OF A SIMPLE PROGRAM

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 Ø 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 (Ø 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 (Ø=left, 1=right, 2=middle) plus 4 times the channel number (Ø-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 (Ø=left, 1=middle, 2=right). The two least significant bits indicate the channel number within the stereo position (Ø 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 Ø. 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 16Ø 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 Ø.

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. Ø to 127 will add a value of Ø 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 3Ø, it will be raised by multiples of 24 until it is 3Ø 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 Ø 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 Ø 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 Ø 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 Ø to 191. The range of the Apple Music][is limited to values from 3Ø 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 Ø 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 3Ø; 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

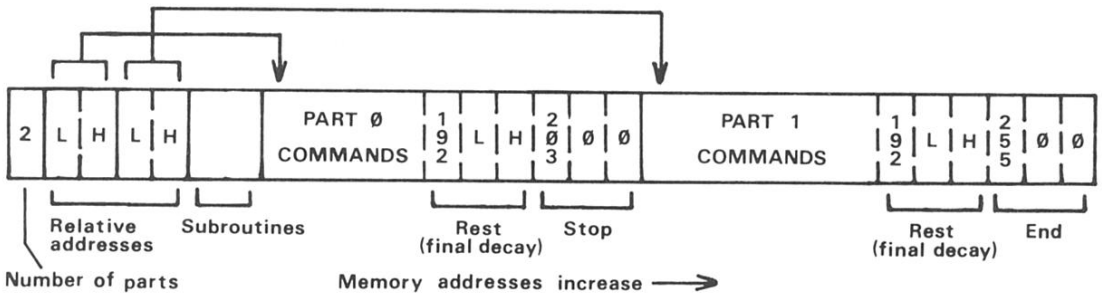
RELATIVE ADDRESSES

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



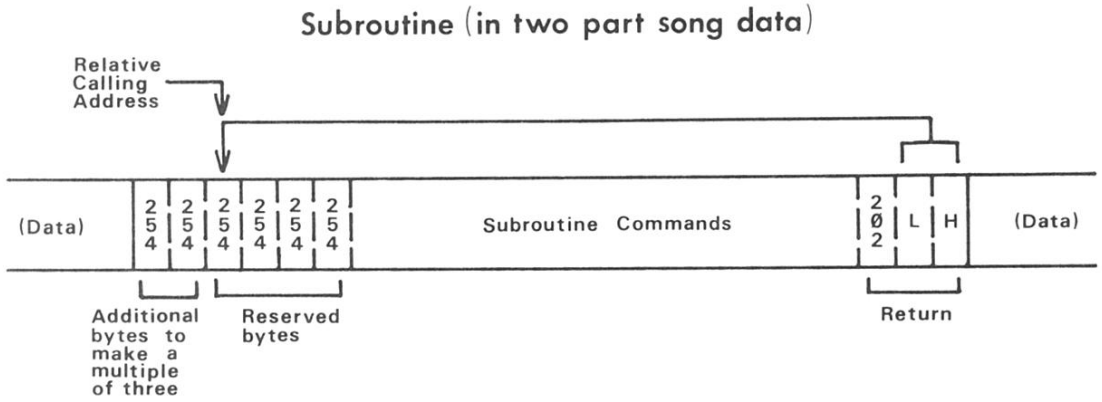
PART DATA

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 preceed 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 preceeded 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 1Ø-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.

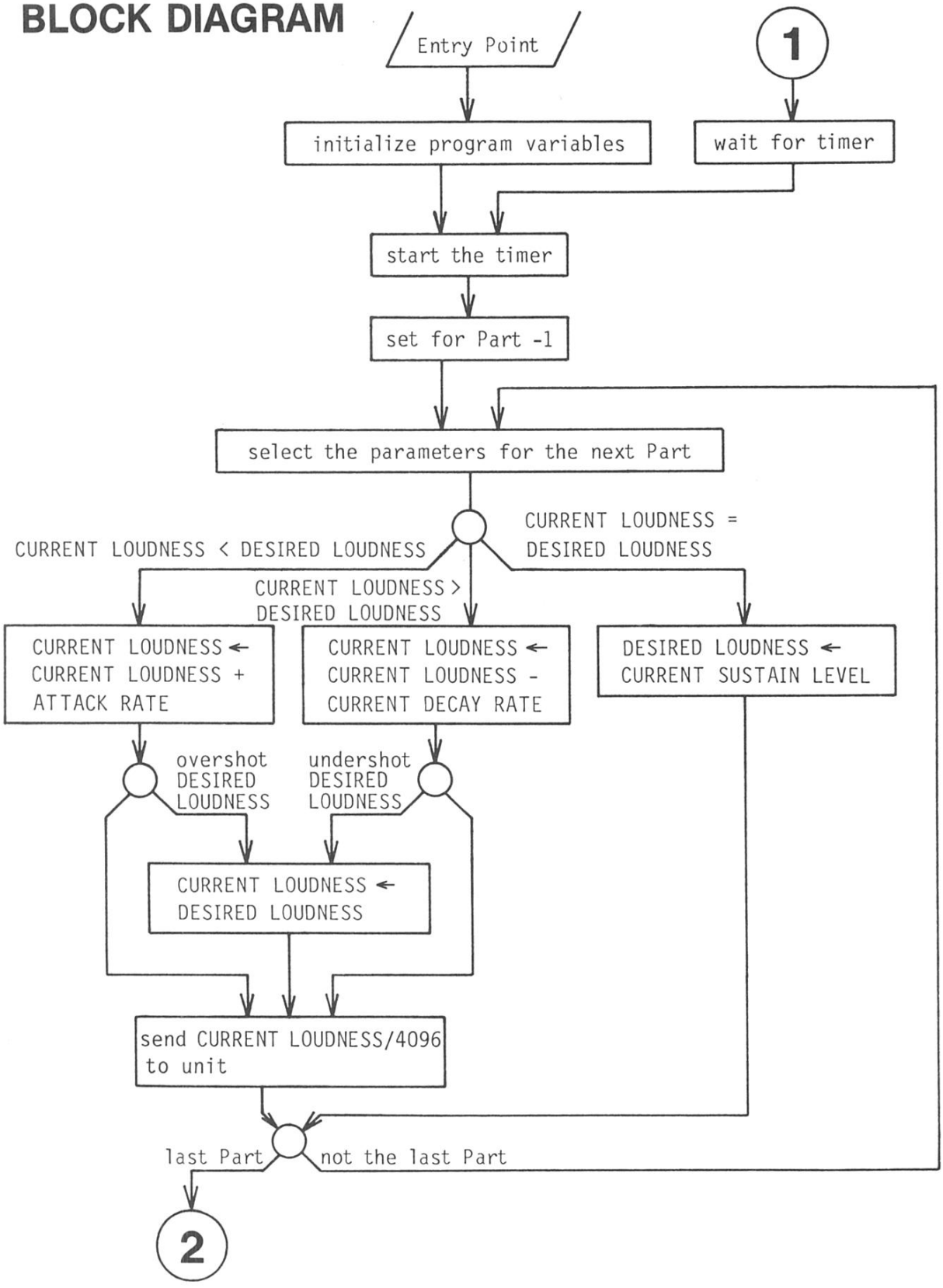
TEMPORARIES

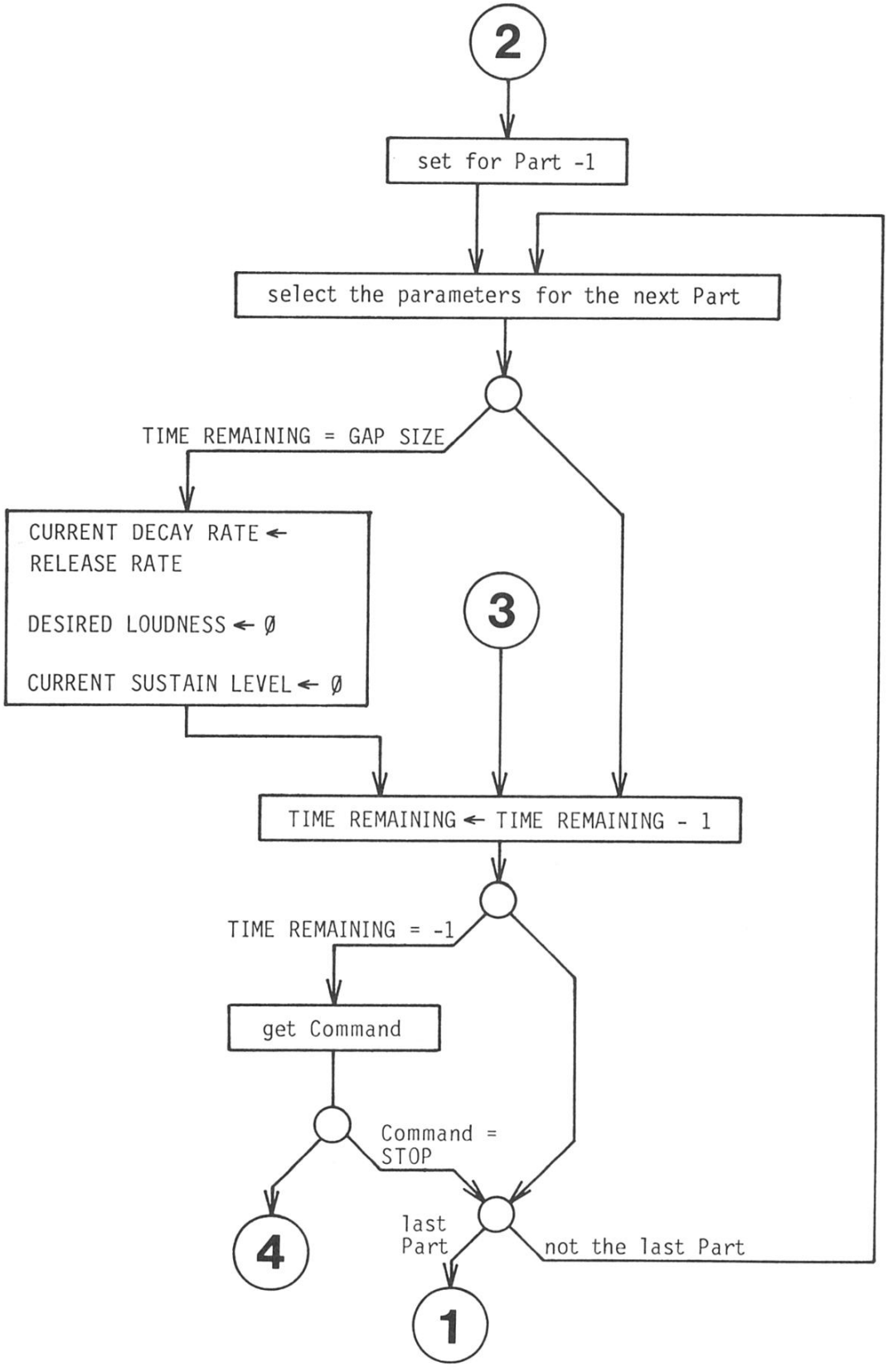
PERFORM uses locations Ø-19 (hex) (6-C and DD-EF for the Applesoft version) for storage of temporary values during execution.

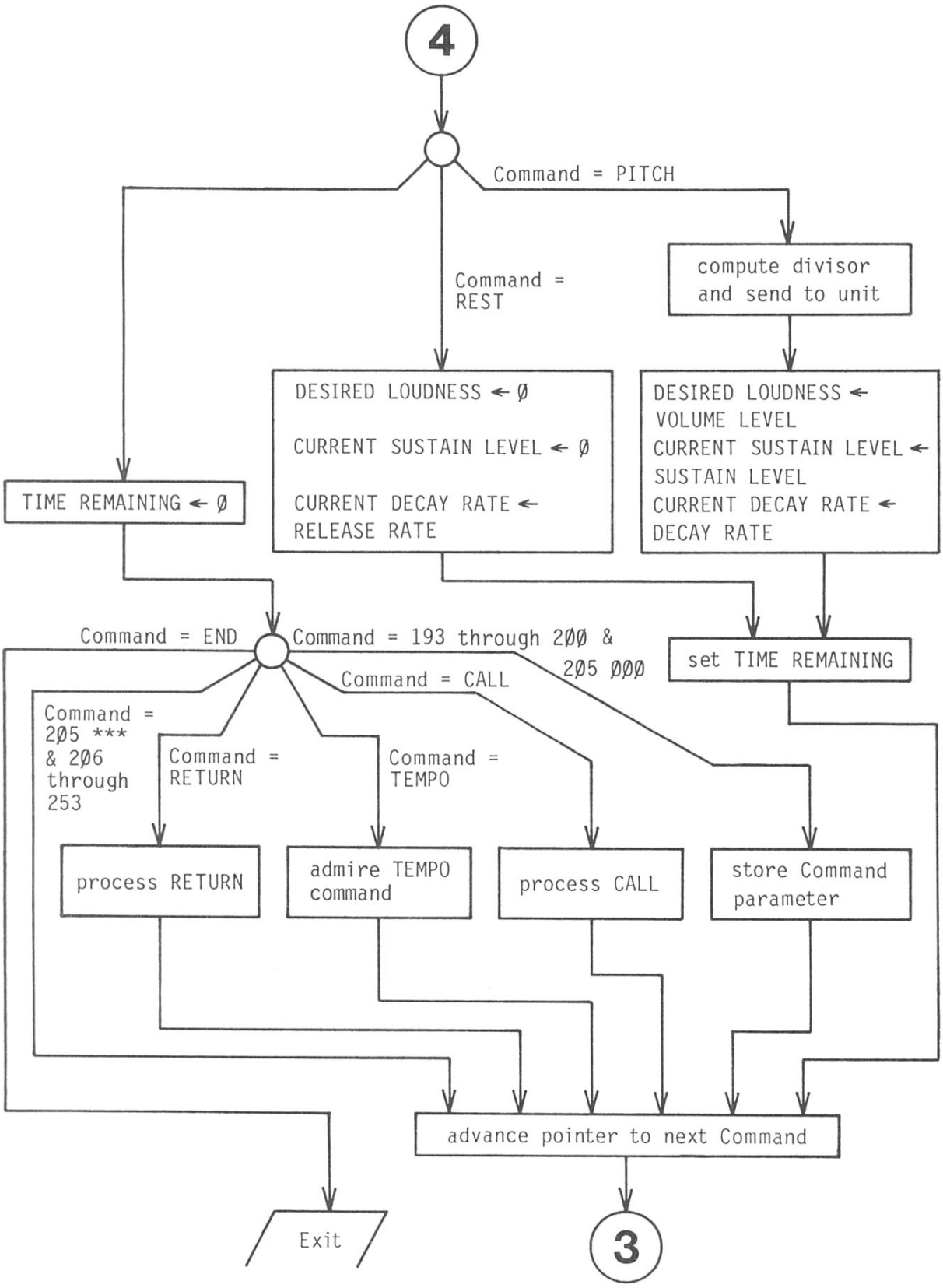
COMMAND NUMBERS

HEX	DECIMAL	COMMAND
Ø-BF	Ø-191	PITCH
CØ	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	2ØØ	CHANNEL NUMBER
C9	2Ø1	CALL
CA	2Ø2	RETURN
CB	2Ø3	STOP
CC	2Ø4	TEMPO
CD ØØ	2Ø5 ØØØ	FUZZ
CD **	2Ø5 ***	(where ** or *** are non-zero) no operation
CE-FD	2Ø6-253	no operation
FE	254	preceeds subroutines, treated as END if found
FF	255	END

BLOCK DIAGRAM




2
set for Part -1
select the parameters for the next Part
TIME REMAINING = GAP SIZE
CURRENT DECAY RATE ←
RELEASE RATE
DESIRED LOUDNESS ← Ø
CURRENT SUSTAIN LEVEL ← Ø
3
TIME REMAINING ← TIME REMAINING - 1
TIME REMAINING = -1
get Command
Command =
STOP
4
last
Part
not the last Part
1


4
Command = PITCH
Command = REST
compute divisor and send to unit
DESIRED LOUDNESS ← Ø
CURRENT SUSTAIN LEVEL ← Ø
CURRENT DECAY RATE ← RELEASE RATE
DESIRED LOUDNESS ← VOLUME LEVEL
CURRENT SUSTAIN LEVEL ← SUSTAIN LEVEL
CURRENT DECAY RATE ← DECAY RATE
TIME REMAINING ← Ø
set TIME REMAINING
Command = END
Command = 193 through 2ØØ & 2Ø5 ØØØ
Command = CALL
Command = 2Ø5 *** & 2Ø6 through 253
Command = RETURN
Command = TEMPO
process RETURN
admire TEMPO command
process CALL
store Command parameter
advance pointer to next Command
Exit
3

6

PROGRAMMING

BARE HANDED

The Apple Music][is programmed by means of 3 "ports". Each port controls three channels (numbered Ø-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 form Ø to 255) to that memory address (using 65Ø2 Assembly Language or BASIC's POKE). Reading from these memory addresses does not affect the synthesizer. The ports are numbered from Ø 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 (Ø=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=Ø 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 (Ø to 3) and V is the volume (Ø 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 2Ø8. This could be done using POKE SLOT*16-16256+P,2Ø8.

The amount of attenuation (from full volume) is approximately 3Ø-2*V dB for values of V from 1 to 15. V=Ø is "off" (infinite attenuation).

FREQUENCY

The frequencies (pitches) 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 6392Ø/D Hz, where D is an integer from 1 to 1Ø23, can be selected. Two bytes must be written. The first byte is computed by the formula 128+32*C+D MOD 16, where C is the channel number (Ø to 2) and D is the frequency divisor (1 to 1Ø23). Using INT instead of MOD, the formula is 128+32*C+D-INT(D/16)*16. The second byte is computed by the formula D/16, or INT(D/16). Both bytes must be written at address SLOT*16-16256+P, and the first byte must be written first.

65Ø2 PROGRAMMING

When programming in 65Ø2 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(Ø)*(2 ↑ (N/X)) where X is the number of notes per octave, F(Ø) is the frequency (pitch) of the lowest note in the octave (in Hz, or cycles per second), and N must be an integer from Ø 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,Ø)*(2 ↑ Q) where F(N,Ø) 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=44Ø Hz tuning). Thus the frequency, F(N,Q), of any piano note is F(N,Q)=27.5*(2 ↑ (Q+N/12)) Hz, where N is the note number from Ø to 11 and Q is the octave number from Ø to 7. (Note: pianos have no notes where N is greater than 3 if Q is 7. N=Ø indicates an A natural pitch.) Therefore the desired divisors for piano notes are: D(N,Q)=INT(6392Ø/(27.5*(2 ↑ (Q+N/12)))+Ø.5). Note that the 12

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

Note that since divisors less than 1 or greater than 1Ø23 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 Ø
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 $8ØØ
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 ØØ Ø8  220            LDA DPNTR         SET SONG
0807 85 ØØ     230            STA /SPNTR         DATA POINTER
0809 AD Ø1 Ø8  240            LDA DPNTR+1
080C 85 Ø1     250            STA /SPNTR+1
080E A0 ØØ     260            LDY #Ø            GET NUMBER
0810 B1 ØØ     270            LDA (SPNTR),Y      OF PARTS
0812 ØA        280            ASL A
0813 85 Ø7     290            STA /PARNUM
0815 A2 ØØ     300            LDX #Ø            SET UP
0817 C8        310    CPYADR  INY                PART POINTERS
0818 B1 ØØ     320            LDA (SPNTR),Y
081A 18        330            CLC
081B 65 ØØ     340            ADC /SPNTR
081D 95 Ø8     350            STA /PARPNT,X
081F E8        360            INX
0820 C8        370            INY
0821 B1 ØØ     380            LDA (SPNTR),Y
0823 65 Ø1     390            ADC /SPNTR+1
0825 95 Ø8     400            STA /PARPNT,X
0827 E8        410            INX
0828 E4 Ø7     420            CPX /PARNUM
082A DØ EB     430            BNE CPYADR
082C 46 Ø7     440            LSR /PARNUM
082E A2 F3     450            LDX #ASIZE*9      CLEAR
0830 A9 ØØ     460            LDA #Ø             PARAMETER AREA
0832 9D FF ØA  470    CLEAR   STA TIME-1,X
0835 CA        480            DEX
0836 DØ FA     490            BNE CLEAR
0838           500    * MAIN EXECUTION LOOP
0838 A5 Ø7     510            LDA /PARNUM       SET UP
```

```
Ø83A  85 Ø2      52Ø            STA /COUNT          PART COUNTER
Ø83C  A2 ØØ      53Ø   MAIN     LDX #Ø
Ø83E  8D 7Ø CØ   54Ø            STA $CØ7Ø           START TIMER
Ø841             55Ø   * ENVELOPE PROCESSING SECTION
Ø841  BD 12 ØB   56Ø   ENVEL    LDA LOUDNS,X        CHECK CL (CURRENT LOUDNESS)
Ø844  38         57Ø            SEC                  AND DL (DESIRED LOUDNESS)
Ø845  FD 16 ØB   58Ø            SBC DESIRE,X
Ø848  85 Ø3      59Ø            STA /TEMP1
Ø84A  BD 13 ØB   6ØØ            LDA LOUDNS+1,X
Ø84D  FD 17 ØB   61Ø            SBC DESIRE+1,X
Ø85Ø  9Ø 12      62Ø            BCC UPLD            BRANCH IF CL<DL
Ø852  Ø5 Ø3      63Ø            ORA /TEMP1
Ø854  DØ 31      64Ø            BNE DWNLD           BRANCH IF CL>DL
Ø856  BD 18 ØB   65Ø            LDA CURSUS,X        CL=DL
Ø859  9D 16 ØB   66Ø            STA DESIRE,X        DL:=CURRENT SUSTAIN LEVEL
Ø85C  BD 19 ØB   67Ø            LDA CURSUS+1,X
Ø85F  9D 17 ØB   68Ø            STA DESIRE+1,X
Ø862  BØ 7Ø      69Ø            BCS NEXTE
Ø864  BD 12 ØB   7ØØ   UPLD     LDA LOUDNS,X        CL:=CL+ATTACK RATE
Ø867  7D Ø6 ØB   71Ø            ADC ATTACK,X
Ø86A  9D 12 ØB   72Ø            STA LOUDNS,X
Ø86D  BD 13 ØB   73Ø            LDA LOUDNS+1,X
Ø87Ø  7D Ø7 ØB   74Ø            ADC ATTACK+1,X
Ø873  9D 13 ØB   75Ø            STA LOUDNS+1,X
Ø876  BØ 31      76Ø            BCS ETHERE          BRANCH IF OVERSHOT DL
Ø878  A8         77Ø            TAY                 COMPARE CL AND DL
Ø879  BD 12 ØB   78Ø            LDA LOUDNS,X
Ø87C  DD 16 ØB   79Ø            CMP DESIRE,X
Ø87F  98         8ØØ            TYA
Ø88Ø  FD 17 ØB   81Ø            SBC DESIRE+1,X
Ø883  9Ø 3C      82Ø            BCC SENDE           DON'T BRANCH IF OVERSHOT DL
Ø885  BØ 22      83Ø            BCS ETHERE
Ø887  BD 12 ØB   84Ø   DWNLD     LDA LOUDNS,X        CL:=CL-CURRENT DECAY RATE
Ø88A  FD 14 ØB   85Ø            SBC DOWN,X
Ø88D  9D 12 ØB   86Ø            STA LOUDNS,X
Ø89Ø  BD 13 ØB   87Ø            LDA LOUDNS+1,X
Ø893  FD 15 ØB   88Ø            SBC DOWN+1,X
Ø896  9D 13 ØB   89Ø            STA LOUDNS+1,X
Ø899  9Ø ØE      9ØØ            BCC ETHERE          BRANCH IF UNDERSHOT DL
Ø89B  BD 16 ØB   91Ø            LDA DESIRE,X        COMPARE CL AND DL
Ø89E  DD 12 ØB   92Ø            CMP LOUDNS,X
Ø8A1  BD 17 ØB   93Ø            LDA DESIRE+1,X
Ø8A4  FD 13 ØB   94Ø            SBC LOUDNS+1,X
Ø8A7  9Ø 18      95Ø            BCC SENDE           DON'T BRANCH IF UNDERSHOT DL
Ø8A9  BD 16 ØB   96Ø   ETHERE    LDA DESIRE,X        CL:=DL
Ø8AC  9D 12 ØB   97Ø            STA LOUDNS,X
Ø8AF  BD 17 ØB   98Ø            LDA DESIRE+1,X
Ø8B2  9D 13 ØB   99Ø            STA LOUDNS+1,X
Ø8B5  BD 18 ØB   1ØØØ           LDA CURSUS,X        DL:=CURRENT SUSTAIN LEVEL
Ø8B8  9D 16 ØB   1Ø1Ø           STA DESIRE,X
Ø8BB  BD 19 ØB   1Ø2Ø           LDA CURSUS+1,X
Ø8BE  9D 17 ØB   1Ø3Ø           STA DESIRE+1,X
Ø8C1  BC 1Ø ØB   1Ø4Ø   SENDE    LDY CHAN,X          SEND LOUDNESS
Ø8C4  BD 13 ØB   1Ø5Ø           LDA LOUDNS+1,X       TO UNIT
```

```
Ø8C7  4A        1Ø6Ø          LSR A
Ø8C8  4A        1Ø7Ø          LSR A
Ø8C9  4A        1Ø8Ø          LSR A
Ø8CA  4A        1Ø9Ø          LSR A
Ø8CB  5D 11 ØB  11ØØ          EOR CHAN+1,X
Ø8CE  1D 1A ØB  111Ø          ORA PERCUS,X
Ø8D1  99 8Ø CØ  112Ø          STA $CØ8Ø,Y
Ø8D4  8A        113Ø NEXTE    TXA                 REPEAT FOR
Ø8D5  18        114Ø          CLC                  NEXT PART
Ø8D6  69 1B     115Ø          ADC #ASIZE
Ø8D8  AA        116Ø          TAX
Ø8D9  C6 Ø2     117Ø          DEC /COUNT
Ø8DB  FØ Ø3     118Ø          BEQ CONT1
Ø8DD  4C 41 Ø8  119Ø          JMP ENVEL
Ø8EØ  A2 ØØ     12ØØ CONT1    LDX #Ø              INITIALIZE PART COUNTER
Ø8E2            121Ø * NOTE DURATION SECTION
Ø8E2  BD ØØ ØB  122Ø LENGTH   LDA TIME,X          COMPARE TIME REMAINING
Ø8E5  DD Ø2 ØB  123Ø          CMP GAP,X            AND GAP SIZE
Ø8E8  DØ 22     124Ø          BNE DECR            BRANCH IF UNEQUAL
Ø8EA  BD Ø1 ØB  125Ø          LDA TIME+1,X
Ø8ED  DD Ø3 ØB  126Ø          CMP GAP+1,X
Ø8FØ  DØ 1A     127Ø          BNE DECR            BRANCH IF UNEQUAL
Ø8F2  BD ØE ØB  128Ø          LDA RELEAS,X        EQUAL; START NOTE
Ø8F5  9D 14 ØB  129Ø          STA DOWN,X           RELEASE
Ø8F8  BD ØF ØB  13ØØ          LDA RELEAS+1,X        CURRENT DECAY RATE:=
Ø8FB  9D 15 ØB  131Ø          STA DOWN+1,X           RELEASE RATE
Ø8FE  A9 ØØ     132Ø          LDA #Ø                DL:=Ø
Ø9ØØ  9D 16 ØB  133Ø          STA DESIRE,X          CURRENT SUSTAIN LEVEL:=Ø
Ø9Ø3  9D 17 ØB  134Ø          STA DESIRE+1,X
Ø9Ø6  9D 18 ØB  135Ø          STA CURSUS,X
Ø9Ø9  9D 19 ØB  136Ø          STA CURSUS+1,X
Ø9ØC  BD ØØ ØB  137Ø DECR     LDA TIME,X          DECREMENT TIME REMAINING
Ø9ØF  DØ Ø8     138Ø          BNE DECR2
Ø911  BD Ø1 ØB  139Ø          LDA TIME+1,X
Ø914  FØ 1B     14ØØ          BEQ PROCES          BRANCH IF NO TIME LEFT
Ø916  DE Ø1 ØB  141Ø DECR1    DEC TIME+1,X
Ø919  DE ØØ ØB  142Ø DECR2    DEC TIME,X
Ø91C  8A        143Ø NEXTL    TXA                 CONTINUE WITH
Ø91D  18        144Ø          CLC                  NEXT PART
Ø91E  69 1B     145Ø          ADC #ASIZE
Ø92Ø  AA        146Ø          TAX
Ø921  E6 Ø2     147Ø          INC /COUNT
Ø923  A5 Ø2     148Ø          LDA /COUNT
Ø925  C5 Ø7     149Ø          CMP /PARNUM
Ø927  DØ B9     15ØØ          BNE LENGTH
Ø929  2C 64 CØ  151Ø WAIT     BIT $CØ64           WAIT FOR TIMER
Ø92C  3Ø FB     152Ø          BMI WAIT
Ø92E  4C 3C Ø8  153Ø          JMP MAIN
Ø931            154Ø * SONG DATA COMMAND PROCESSING SECTION
Ø931  8A        155Ø PROCES   TXA
Ø932  A8        156Ø          TAY
Ø933  A5 Ø2     157Ø          LDA /COUNT
Ø935  ØA        158Ø          ASL A
Ø936  AA        159Ø          TAX
```

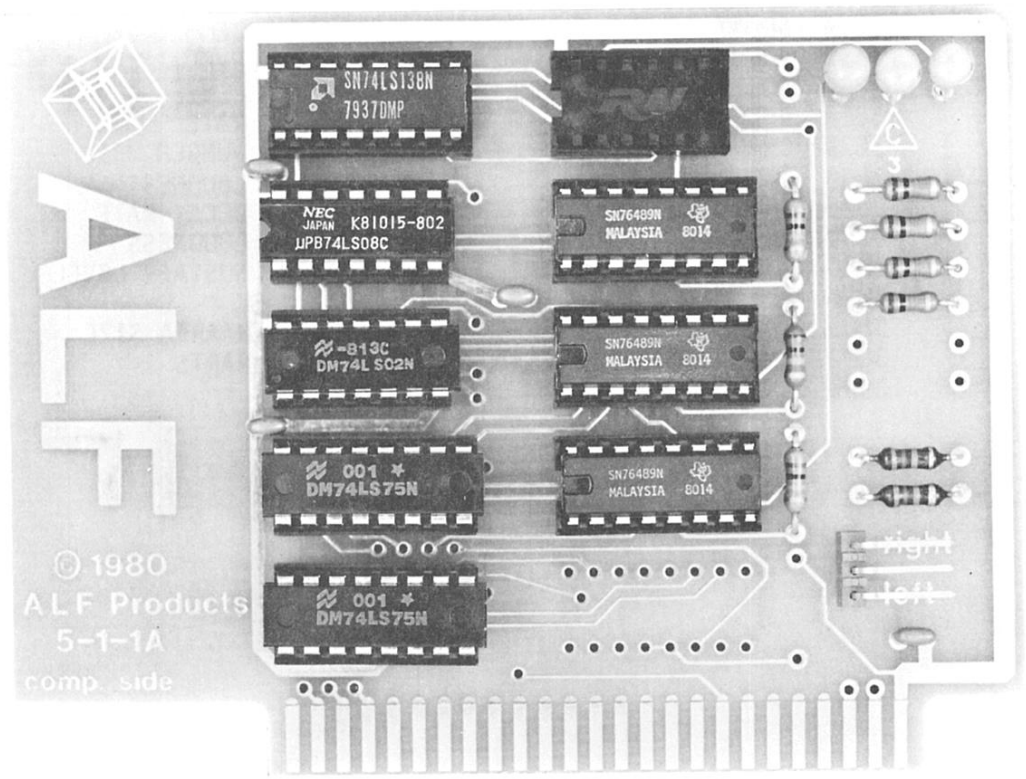
```
Ø937  A1 Ø8     16ØØ           LDA (PARPNT,X)     GET COMMAND TYPE
Ø939  C9 CB     161Ø           CMP #2Ø3
Ø93B  DØ Ø4     162Ø           BNE NOSTOP         BRANCH IF NOT "STOP"
Ø93D            163Ø  * PROCESS STOP COMMAND
Ø93D  98        164Ø           TYA                DO NOTHING
Ø93E  AA        165Ø           TAX
Ø93F  BØ D5     166Ø           BCS DECR1
Ø941  F6 Ø8     167Ø  NOSTOP   INC /PARPNT,X
Ø943  DØ Ø2     168Ø           BNE NOCAR1
Ø945  F6 Ø9     169Ø           INC /PARPNT+1,X
Ø947  C9 CØ     17ØØ  NOCAR1   CMP #192
Ø949  BØ 65     171Ø           BCS NPITCH         BRANCH IF NOT "PITCH"
Ø94B            172Ø  * PROCESS PITCH COMMAND
Ø94B  39 Ø5 ØB  173Ø           AND TRANS+1,Y
Ø94E  79 Ø4 ØB  174Ø           ADC TRANS,Y
Ø951  38        175Ø           SEC
Ø952  E9 1E     176Ø           SBC #3Ø
Ø954  BØ Ø4     177Ø           BCS PITCH2         MOVE THE LOWEST 3Ø
Ø956  69 18     178Ø  PITCH1   ADC #24             PITCHS UP AN OCTAVE
Ø958  9Ø FC     179Ø           BCC PITCH1           OR TWO
Ø95A  86 Ø4     18ØØ  PITCH2   STX /TEMP2         COMPUTE DIVISOR
Ø95C  A2 ØØ     181Ø           LDX #Ø             DIVIDE PITCH BY 24
Ø95E  C9 18     182Ø  DIV      CMP #24
Ø96Ø  9Ø Ø5     183Ø           BCC DIV1
Ø962  E9 18     184Ø           SBC #24
Ø964  E8        185Ø           INX
Ø965  DØ F7     186Ø           BNE DIV
Ø967  86 Ø3     187Ø  DIV1     STX /TEMP1
Ø969  ØA        188Ø           ASL A              LOOK UP
Ø96A  AA        189Ø           TAX                 SUB-OCTAVE DIVISOR
Ø96B  BD 9C ØA  19ØØ           LDA TABLE+1,X
Ø96E  85 Ø5     191Ø           STA /TEMP3
Ø97Ø  BD 9B ØA  192Ø           LDA TABLE,X
Ø973  A6 Ø3     193Ø           LDX /TEMP1
Ø975  CA        194Ø  OCTAVE   DEX                DIVIDE DIVISOR
Ø976  3Ø Ø6     195Ø           BMI ROUND           TO RIGHT OCTAVE
Ø978  46 Ø5     196Ø           LSR /TEMP3
Ø97A  6A        197Ø           ROR A
Ø97B  4C 75 Ø9  198Ø           JMP OCTAVE
Ø97E  69 Ø8     199Ø  ROUND    ADC #8             ROUND RESULT
Ø98Ø  9Ø Ø2     2ØØØ           BCC SENDP
Ø982  E6 Ø5     2Ø1Ø           INC /TEMP3
Ø984  BE 1Ø ØB  2Ø2Ø  SENDP    LDX CHAN,Y         SEND PITCH TO UNIT
Ø987  Ø9 ØF     2Ø3Ø           ORA #$F
Ø989  4A        2Ø4Ø           LSR A
Ø98A  6A        2Ø5Ø           ROR A
Ø98B  6A        2Ø6Ø           ROR A
Ø98C  6A        2Ø7Ø           ROR A
Ø98D  39 11 ØB  2Ø8Ø           AND CHAN+1,Y
Ø99Ø  9D 8Ø CØ  2Ø9Ø           STA $CØ8Ø,X
Ø993  84 Ø3     21ØØ           STY /TEMP1
Ø995  EA        211Ø           NOP
Ø996  EA        212Ø           NOP
Ø997  EA        213Ø           NOP
```

```
Ø998  EA        214Ø           NOP
Ø999  A5 Ø5     215Ø           LDA /TEMP3
Ø99B  9D 8Ø CØ  216Ø           STA $CØ8Ø,X
Ø99E  A2 Ø6     217Ø           LDX #6              START "ADSR" CYCLE
Ø9AØ  B9 Ø8 ØB  218Ø  CYCLE    LDA DECAY,Y
Ø9A3  99 14 ØB  219Ø           STA DOWN,Y
Ø9A6  C8        22ØØ           INY
Ø9A7  CA        221Ø           DEX
Ø9A8  DØ F6     222Ø           BNE CYCLE
Ø9AA  A6 Ø4     223Ø           LDX /TEMP2
Ø9AC  A4 Ø3     224Ø           LDY /TEMP1          STORE NOTE TIME
Ø9AE  BØ 22     225Ø           BCS STORD1
Ø9BØ  DØ 3B     226Ø  NPITCH   BNE NREST           BRANCH IF NOT "REST"
Ø9B2            227Ø  * PROCESS REST COMMAND
Ø9B2  B9 ØE ØB  228Ø           LDA RELEAS,Y        DO A "RELEASE"
Ø9B5  99 14 ØB  229Ø           STA DOWN,Y
Ø9B8  B9 ØF ØB  23ØØ           LDA RELEAS+1,Y
Ø9BB  99 15 ØB  231Ø           STA DOWN+1,Y
Ø9BE  A9 ØØ     232Ø           LDA #Ø
Ø9CØ  99 16 ØB  233Ø           STA DESIRE,Y
Ø9C3  99 17 ØB  234Ø           STA DESIRE+1,Y
Ø9C6  99 18 ØB  235Ø           STA CURSUS,Y
Ø9C9  99 19 ØB  236Ø           STA CURSUS+1,Y
Ø9CC  18        237Ø           CLC
Ø9CD  84 Ø3     238Ø  STORD    STY /TEMP1          STORE PARAMETER
Ø9CF  65 Ø3     239Ø           ADC /TEMP1           IN PARAMETER AREA
Ø9D1  A8        24ØØ           TAY
Ø9D2  A1 Ø8     241Ø  STORD1   LDA (PARPNT,X)
Ø9D4  99 ØØ ØB  242Ø           STA TIME,Y
Ø9D7  F6 Ø8     243Ø           INC /PARPNT,X
Ø9D9  DØ Ø2     244Ø           BNE NOCAR2
Ø9DB  F6 Ø9     245Ø           INC /PARPNT+1,X
Ø9DD  A1 Ø8     246Ø  NOCAR2   LDA (PARPNT,X)
Ø9DF  99 Ø1 ØB  247Ø           STA TIME+1,Y
Ø9E2  F6 Ø8     248Ø  FIXUP    INC /PARPNT,X
Ø9E4  DØ Ø2     249Ø           BNE NOCAR3
Ø9E6  F6 Ø9     25ØØ           INC /PARPNT+1,X
Ø9E8  A6 Ø3     251Ø  NOCAR3   LDX /TEMP1
Ø9EA  4C ØC Ø9  252Ø           JMP DECR
Ø9ED  85 Ø3     253Ø  NREST    STA /TEMP1          SET SO COMMAND
Ø9EF  A9 ØØ     254Ø           LDA #Ø               TAKES ZERO TIME
Ø9F1  99 ØØ ØB  255Ø           STA TIME,Y
Ø9F4  99 Ø1 ØB  256Ø           STA TIME+1,Y
Ø9F7  A5 Ø3     257Ø           LDA /TEMP1
Ø9F9  C9 C8     258Ø           CMP #2ØØ
Ø9FB  BØ Ø5     259Ø           BCS NSTORE          BRANCH IF NOT A
Ø9FD  E9 BF     26ØØ           SBC #191             STORED COMMAND
Ø9FF  ØA        261Ø           ASL A
ØAØØ  DØ CB     262Ø           BNE STORD
ØAØ2  84 Ø3     263Ø  NSTORE   STY /TEMP1
ØAØ4  DØ 13     264Ø           BNE NOCHAN          BRANCH IF NOT "CHANNEL"
ØAØ6            265Ø  * PROCESS CHANNEL COMMAND
ØAØ6  A1 Ø8     266Ø           LDA (PARPNT,X)
ØAØ8  29 F3     267Ø           AND #$F3
```

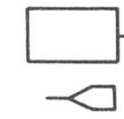
```
ØA0A  99 1Ø ØB  268Ø           STA CHAN,Y
ØA0D  A1 Ø8     269Ø           LDA (PARPNT,X)
ØA0F  ØA        27ØØ           ASL A
ØA1Ø  ØA        271Ø           ASL A
ØA11  ØA        272Ø           ASL A
ØA12  Ø9 9F     273Ø           ORA #$9F
ØA14  99 11 ØB  274Ø           STA CHAN+1,Y
ØA17  DØ 76     275Ø           BNE NOP
ØA19  C9 CA     276Ø  NOCHAN   CMP #2Ø2
ØA1B  BØ 33     277Ø           BCS NOCALL        BRANCH IF NOT A "CALL"
ØA1D            278Ø  * PROCESS CALL COMMAND
ØA1D  A1 Ø8     279Ø           LDA (PARPNT,X)    COMPUTE CALLED ADDRESS
ØA1F  65 ØØ     28ØØ           ADC /SPNTR
ØA21  85 Ø5     281Ø           STA /TEMP3
ØA23  F6 Ø8     282Ø           INC /PARPNT,X
ØA25  DØ Ø2     283Ø           BNE NOCAR4
ØA27  F6 Ø9     284Ø           INC /PARPNT+1,X
ØA29  A1 Ø8     285Ø  NOCAR4   LDA (PARPNT,X)
ØA2B  65 Ø1     286Ø           ADC /SPNTR+1
ØA2D  85 Ø6     287Ø           STA /TEMP3+1
ØA2F  8A        288Ø           TXA               STORE RETURN ADDRESS
ØA3Ø  A8        289Ø           TAY
ØA31  B5 Ø8     29ØØ           LDA /PARPNT,X
ØA33  69 Ø1     291Ø           ADC #1
ØA35  91 Ø5     292Ø           STA (TEMP3),Y
ØA37  C8        293Ø           INY
ØA38  B5 Ø9     294Ø           LDA /PARPNT+1,X
ØA3A  69 ØØ     295Ø           ADC #Ø
ØA3C  91 Ø5     296Ø           STA (TEMP3),Y
ØA3E  A5 Ø7     297Ø           LDA /PARNUM       ADVANCE CALLING
ØA4Ø  ØA        298Ø           ASL A              ADDRESS OVER
ØA41  65 Ø5     299Ø           ADC /TEMP3          RETURN ADDRESSES
ØA43  95 Ø8     3ØØØ           STA /PARPNT,X
ØA45  A5 Ø6     3Ø1Ø           LDA /TEMP3+1
ØA47  69 ØØ     3Ø2Ø           ADC #Ø
ØA49  95 Ø9     3Ø3Ø           STA /PARPNT+1,X
ØA4B  A6 Ø3     3Ø4Ø           LDX /TEMP1
ØA4D  4C ØC Ø9  3Ø5Ø           JMP DECR
ØA5Ø  DØ 23     3Ø6Ø  NOCALL   BNE NORET         BRANCH IF NOT "RETURN"
ØA52            3Ø7Ø  * PROCESS RETURN COMMAND
ØA52  18        3Ø8Ø           CLC
ØA53  A1 Ø8     3Ø9Ø           LDA (PARPNT,X)    COMPUTE RETURN ADDRESS
ØA55  65 ØØ     31ØØ           ADC /SPNTR         ADDRESS
ØA57  85 Ø5     311Ø           STA /TEMP3
ØA59  F6 Ø8     312Ø           INC /PARPNT,X
ØA5B  DØ Ø2     313Ø           BNE NOCAR5
ØA5D  F6 Ø9     314Ø           INC /PARPNT+1,X
ØA5F  A1 Ø8     315Ø  NOCAR5   LDA (PARPNT,X)
ØA61  65 Ø1     316Ø           ADC /SPNTR+1
ØA63  85 Ø6     317Ø           STA /TEMP3+1
ØA65  8A        318Ø           TXA               GO TO
ØA66  A8        319Ø           TAY                RETURN ADDRESS
ØA67  B1 Ø5     32ØØ           LDA (TEMP3),Y
ØA69  95 Ø8     321Ø           STA /PARPNT,X
```

```
ØA6B  C8          322Ø           INY
ØA6C  B1 Ø5       323Ø           LDA (TEMP3),Y
ØA6E  95 Ø9       324Ø           STA /PARPNT+1,X
ØA7Ø  A6 Ø3       325Ø           LDX /TEMP1
ØA72  4C ØC Ø9    326Ø           JMP DECR
ØA75  C9 CD       327Ø NORET     CMP #2Ø5
ØA77  DØ 12       328Ø           BNE NOPERC          BRANCH IF NOT "FUZZ"
ØA79              329Ø * PROCESS FUZZ COMMAND
ØA79  A1 Ø8       33ØØ           LDA (PARPNT,X)
ØA7B  DØ 12       331Ø           BNE NOP
ØA7D  F6 Ø8       332Ø           INC /PARPNT,X
ØA7F  DØ Ø2       333Ø           BNE NOCAR7
ØA81  F6 Ø9       334Ø           INC /PARPNT+1,X
ØA83  A1 Ø8       335Ø NOCAR7    LDA (PARPNT,X)
ØA85  99 1A ØB    336Ø           STA PERCUS,Y
ØA88  4C E2 Ø9    337Ø           JMP FIXUP
ØA8B  C9 FE       338Ø NOPERC    CMP #254
ØA8D  BØ Ø9       339Ø           BCS END             BRANCH IF NOT A "NOP"
ØA8F  F6 Ø8       34ØØ NOP       INC /PARPNT,X
ØA91  DØ Ø2       341Ø           BNE NOCAR6
ØA93  F6 Ø9       342Ø           INC /PARPNT+1,X
ØA95  4C E2 Ø9    343Ø NOCAR6    JMP FIXUP
ØA98  68          344Ø END       PLA                 "PROCESS" END COMMAND
ØA99  AA          345Ø           TAX                 RESTORE X
ØA9A  6Ø          346Ø           RTS                  AND RETURN
ØA9B              347Ø * SUB-OCTAVE DIVISOR TABLE
ØA9B  1Ø 3D       348Ø TABLE     DEF 15632
ØA9D  5Ø 3B       349Ø           DEF 15184
ØA9F  AØ 39       35ØØ           DEF 14752
ØAA1  ØØ 38       351Ø           DEF 14336
ØAA3  7Ø 36       352Ø           DEF 13936
ØAA5  EØ 34       353Ø           DEF 13536
ØAA7  6Ø 33       354Ø           DEF 13152
ØAA9  EØ 31       355Ø           DEF 12768
ØAAB  8Ø 3Ø       356Ø           DEF 12416
ØAAD  2Ø 2F       357Ø           DEF 12Ø64
ØAAF  CØ 2D       358Ø           DEF 11712
ØAB1  7Ø 2C       359Ø           DEF 11376
ØAB3  3Ø 2B       36ØØ           DEF 11Ø56
ØAB5  FØ 29       361Ø           DEF 1Ø736
ØAB7  CØ 28       362Ø           DEF 1Ø432
ØAB9  AØ 27       363Ø           DEF 1Ø144
ØABB  8Ø 26       364Ø           DEF 9856
ØABD  6Ø 25       365Ø           DEF 9568
ØABF  5Ø 24       366Ø           DEF 9296
ØAC1  5Ø 23       367Ø           DEF 9Ø4Ø
ØAC3  4Ø 22       368Ø           DEF 8768
ØAC5  5Ø 21       369Ø           DEF 8528
ØAC7  6Ø 2Ø       37ØØ           DEF 8288
ØAC9  7Ø 1F       371Ø           DEF 8Ø48
ØACB              372Ø * COMMAND PARAMETER AREA
ØBØØ              373Ø           ORG $BØØ
ØBØØ              374Ø TIME      BSS 2               TIME REMAINING
ØBØ2              375Ø GAP       BSS 2               GAP SIZE
```

```
ØBØ4            376Ø  TRANS    BSS 2          TRANSPOSE VALUE
ØBØ6            377Ø  ATTACK   BSS 2          ATTACK RATE
ØBØ8            378Ø  DECAY    BSS 6          DECAY RATE
ØBØA            379Ø  VOLUME   EQU DECAY+2     VOLUME LEVEL
ØBØC            38ØØ  SUSTAN   EQU VOLUME+2   SUSTAIN LEVEL
ØBØE            381Ø  RELEAS   BSS 2          RELEASE RATE
ØB1Ø            382Ø  CHAN     BSS 2          CHANNEL NUMBER
ØB12            383Ø  LOUDNS   BSS 2          CURRENT LOUDNESS
ØB14            384Ø  DOWN     BSS 6          CURRENT DECAY RATE
ØB16            385Ø  DESIRE   EQU DOWN+2     DESIRED LOUDNESS
ØB18            386Ø  CURSUS   EQU DESIRE+2   CURRENT SUSTAIN LEVEL
ØB1A            387Ø  PERCUS   BSS 1          FUZZ MASK
ØØ1B            388Ø  ASIZE    EQU *-TIME     PARAMETER AREA SIZE
ØB1B            389Ø           BSS ASIZE*8    OTHER 8 PARTS
ØBF3            39ØØ           END
```



SCHEMATIC TERMINALS



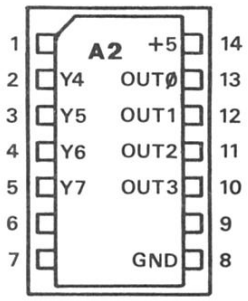
Connection to pin on Apple II Peripheral I/O Bus

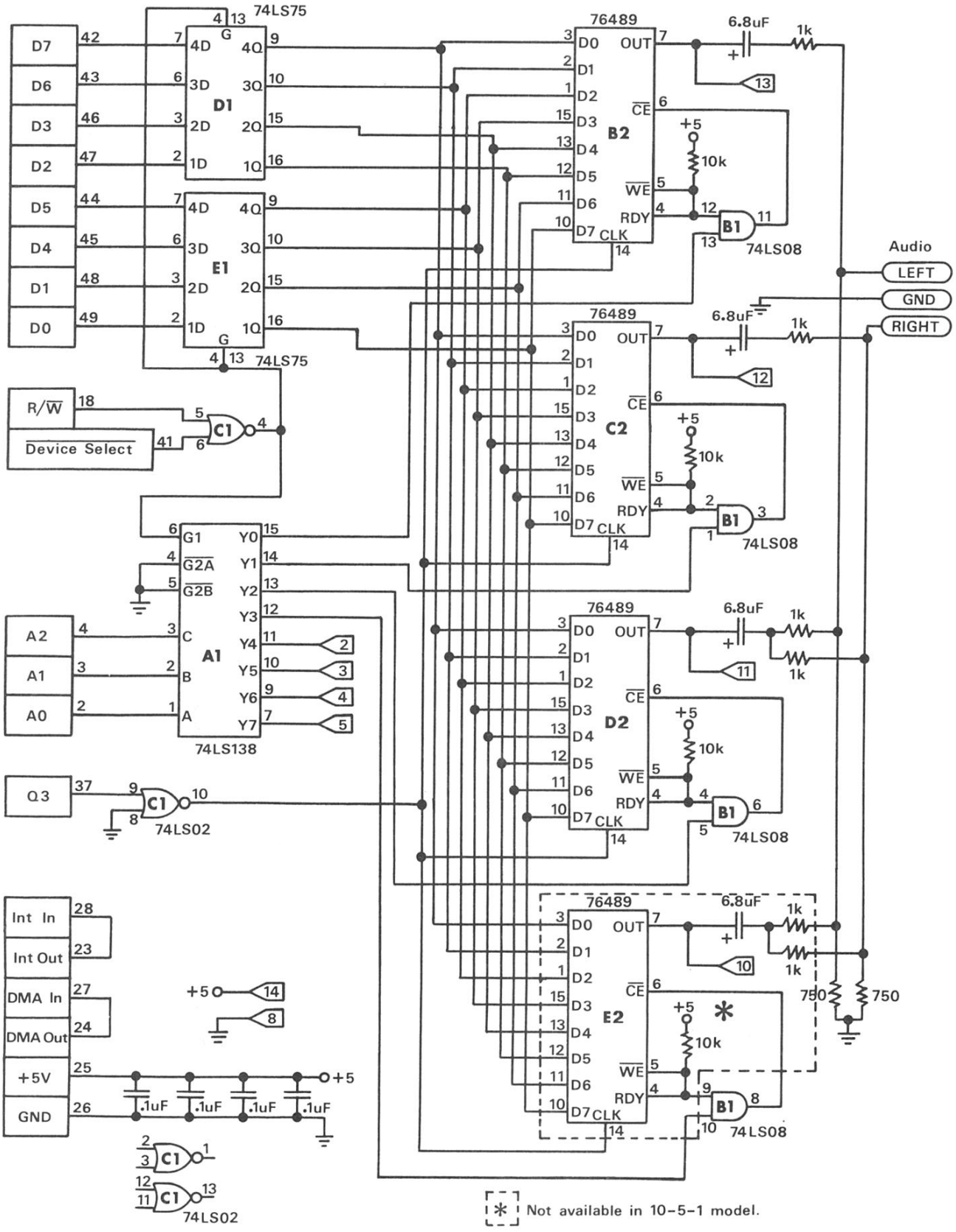
Connection to pin on Access Socket **A2**

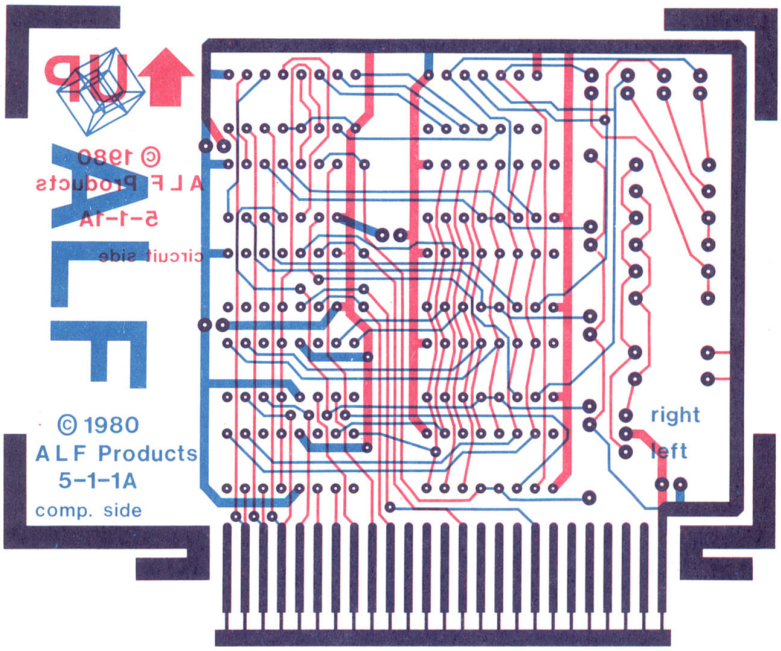
Connection to Audio Out molex pin

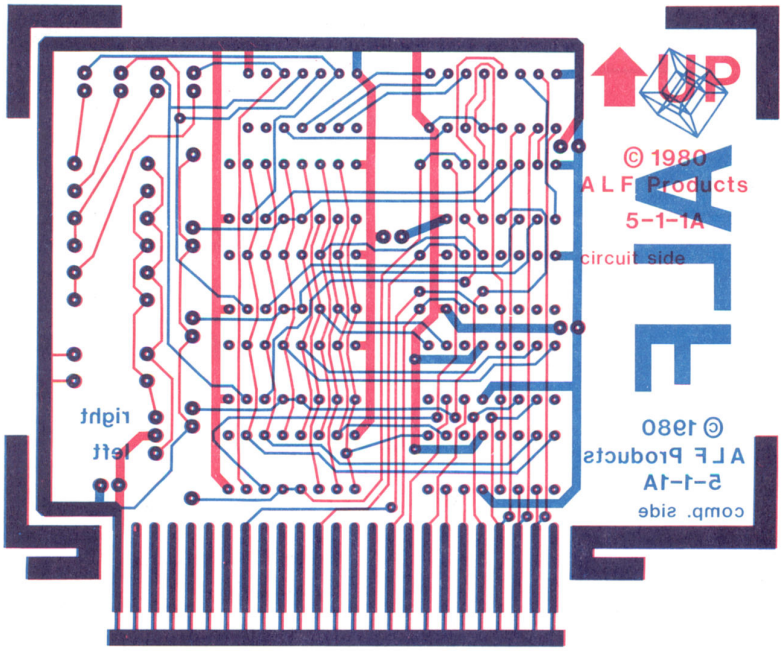
Connection to power bus

ACCESS SOCKET




74LS75
D7
D6
D3
D2
D5
D4
D1
D0
D1
E1
74LS75
R/W
Device Select
C1
74LS138
A2
A1
A0
A1
Q3
74LS02
Int In
Int Out
DMA In
DMA Out
+5V
GND
.1uF
C1
74LS02
76489
B2
C2
D2
E2
6.8uF
1k
10k
+5
CE
WE
RDY
CLK
B1
74LS08
Audio
LEFT
GND
RIGHT
750
* Not available in 10-5-1 model.


© 1980
A L F Products
5-1-1A
comp. side
right
left


© 1980
A L F Products
5-1-1A
circuit side
right
left

INDEX