MUSIC CARD
MC1 & MC16

ALF

The

Music Card Owner's Manual

Complete Instructions

for the

10-5-1 MC1
&
10-5-16 MC16
Music Cards

"the amazing thing about a Dancing Bear
is not how well he Dances;
but that he can
Dance At All"

ALF Products Inc.
1315F Nelson Street
Denver, CO 80215
U.S.A.

Part Number 11-1-1C

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

MUSIC CARD MC1 & MUSIC CARD MC16
FULL 1 YEAR WARRANTY

ALF Products Inc. warrants that computer programs will function as described in their associated owner's manuals, and that all other items will be free of defects in material and workmanship. ALF will correct any fault in a computer program (or its manual, or both) or repair or replace (at ALF's choice) any defective item free of charge for one year from the date of sale by ALF.

To obtain warranty service, you must contact ALF at 1315F Nelson Street, Denver, Colorado 80215 or (303)·234-0871 for a service address. You must send the complete product, proof of purchase date, and a detailed description of the difficulty to the service address. You pay for shipment to ALF, ALF pays for shipment back.

Any alteration of the product serial number voids this warranty. This warranty covers only ALF's products, so where local laws permit **ALF will not be liable for consequential damages.**

Ask your state government for details on their "implied warranty" which also covers this product.

The following statements, which shed no new meaning on this warranty, are required by Federal Trade Commission regulations and are meant to simplify warranty language: **"Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you."** and **"This warranty gives you specific legal rights, and you may also have other rights which vary from state to state."**

CONGRATULATIONS!

ALF's "MC" series of music cards is recognized world wide as the standard of excellence for Apple-compatible music peripherals. The legendary MC16, originally introduced in 1979 as the "Apple Music Synthesizer", was the first music card available for the Apple. Many software companies have chosen the MC series as the only Apple music card supported by their programs. The unmatched signal quality and fine craftsmanship of the MC16 is demanded by discriminating music educators at prestigious universities. Even the most demanding perfectionists have praised the music cards' versatility, ease of use, and excellent human interface.

Your choice of an ALF "MC" series music card reflects your appreciation of advanced technology in a musical instrument. With your purchase, you join the elite group of ALF users — thousands of intelligent, sophisticated computer owners with a variety of interests. Your music card places a new dimension of music at your fingertips, and allows you to explore new regions of your own creativity.

We've enjoyed creating this music card for you, and we sincerely hope it will continue to give you enjoyment for years to come. Thank you for choosing ALF.

Project engineers: Rick Harman, John Ridges, Philip Tubb.
Software design: Tim Gill, John Ridges, Forrest Thiessen, Philip Tubb.
Manual written by: Philip Tubb.
Index by: Greg Bloom.
Manual graphics by: Rick Harman, Mary Otis.
Automatic graphics by: John Ridges, Philip Tubb.
Photos: Chuck Renstrom.
Printed by: Generation Printing.

CONTENTS

1. INSTALLATION

1-1 MC16 cable connection
1-2 Installation
1-4 Operating tips
1-4 Problem checklist
1-5 Copyright laws

2. ENTRY

2-1 Entering a simple song
2-12 Correcting mistakes
2-15 Entering rests
2-16 Subroutines
2-19 Loading and saving songs
2-20 Adjusting the tempo
2-21 Envelopes
2-27 Beaming
2-28 Sample song breakdown
2-29 Summary of commands
2-29 Type 1 commands
2-29 Note duration
2-29 . 3
2-30 ♯ ♭ ♮
2-30 → ←
2-30 INS
2-30 ♮
2-30 FP
2-31 GOTO
2-31 LENGTH
2-31 MEASURE
2-31 PART
2-31 PLAY
2-32 SAVE
2-32 ***DISK
2-32 Type 2 commands
2-32 DEL
2-33 DELETE
2-33 EDIT
2-33 LOAD
2-33 NEW
2-34 SPEED
2-34 STEREO
2-35 SUBROUTINE
2-35 Type 3 commands
2-35 KEY
2-36 QUARTER
2-36 TIME
2-36 Type 4 commands
2-37 REST
2-37 PADDLE 1
2-37 TIE
2-37 ATTACK
2-38 CALL
2-38 DECAY
2-38 FUZZ
2-38 GAP
2-38 POKE
2-39 RATE
2-39 RELEASE
2-39 SUSTAIN
2-39 TEMPO
2-39 TRANSPOSE
2-39 VOLUME
2-40 Tips
2-40 Partial starting measure
2-40 Rests at the end of parts
2-40 Speed/rate settings
2-41 Backup
2-41 Transpose
2-41 Copying songs without ENTRY
2-42 Reset
2-42 Integer/Applesoft switch
2-42 MC1 range
2-42 MC16 parts limit
2-42 Song data format
2-43 ENTRY2

3. ENVELOPE

3-1 The ENVELOPE program
3-3 Commands
3-3 Line 10

4. PROCESS/MLIST

4-1 PROCESS
4-1 LOAD
4-1 DELETE
4-2 CHANGE
4-2 STATUS
4-3 WIDTH
4-3 PR#
4-4 AUXILIARY
4-4 APPEND
4-5 SAVE
4-6 BLOAD
4-6 BSAVE
4-6 Rate
4-7 Commands
4-7 Line 10
4-8 MLIST
4-8 LOAD
4-8 LIST
4-9 PR#
4-9 WIDTH
4-9 FWIDTH
4-9 FP
4-9 Control-S
4-10 Sample listing
4-12 Commands
4-12 Rate

5. PLAY/DISCO/HUSTLE

5-1 PLAY
5-1 LOAD
5-1 PLAY
5-1 STOP
5-1 FP
5-1 Reset
5-2 Album files
5-2 DISCO
5-3 HUSTLE
5-3 Playing the album

6. PROGRAMMING THE MC16 WITH CHROMA

6-1 Initializer
6-3 Partial initializer
6-3 Chroma
6-5 Pulse
6-6 Chroma example
6-6 Pulse example

7. PROGRAMMING THE MC1 BARE HANDED

7-1 Initialization
7-1 Volume
7-1 Frequency
7-2 6502 programming
7-2 Divisor calculation

8. PROGRAMMING THE MC16 BARE HANDED

8-1 Ports
8-3 Divisor calculation
8-4 Tuning

9. PROGRAMMING WITH PERFORM (& FLASH)

9-1 An example
9-2 Music card initialization
9-3 Song configuration
9-4 Reading the "initial speed"
9-4 Sample PERFORM session
9-6 FLASH
9-7 Block diagram
9-10 Technical
9-10 Type A commands
9-10 CHANNEL NUMBER
9-10 CALL
9-11 RETURN
9-11 STOP
9-11 END
9-11 Type B commands
9-11 TRANSPOSE
9-12 FUZZ
9-12 GAP SIZE
9-12 ATTACK RATE, DECAY RATE, VOLUME LEVEL, SUSTAIN LEVEL, RELEASE RATE
9-12 Type C commands
9-12 PITCH
9-13 REST
9-13 Song data
9-14 Rate command
9-15 Temporaries
9-15 Command numbers

10. MC1 TECHNICAL

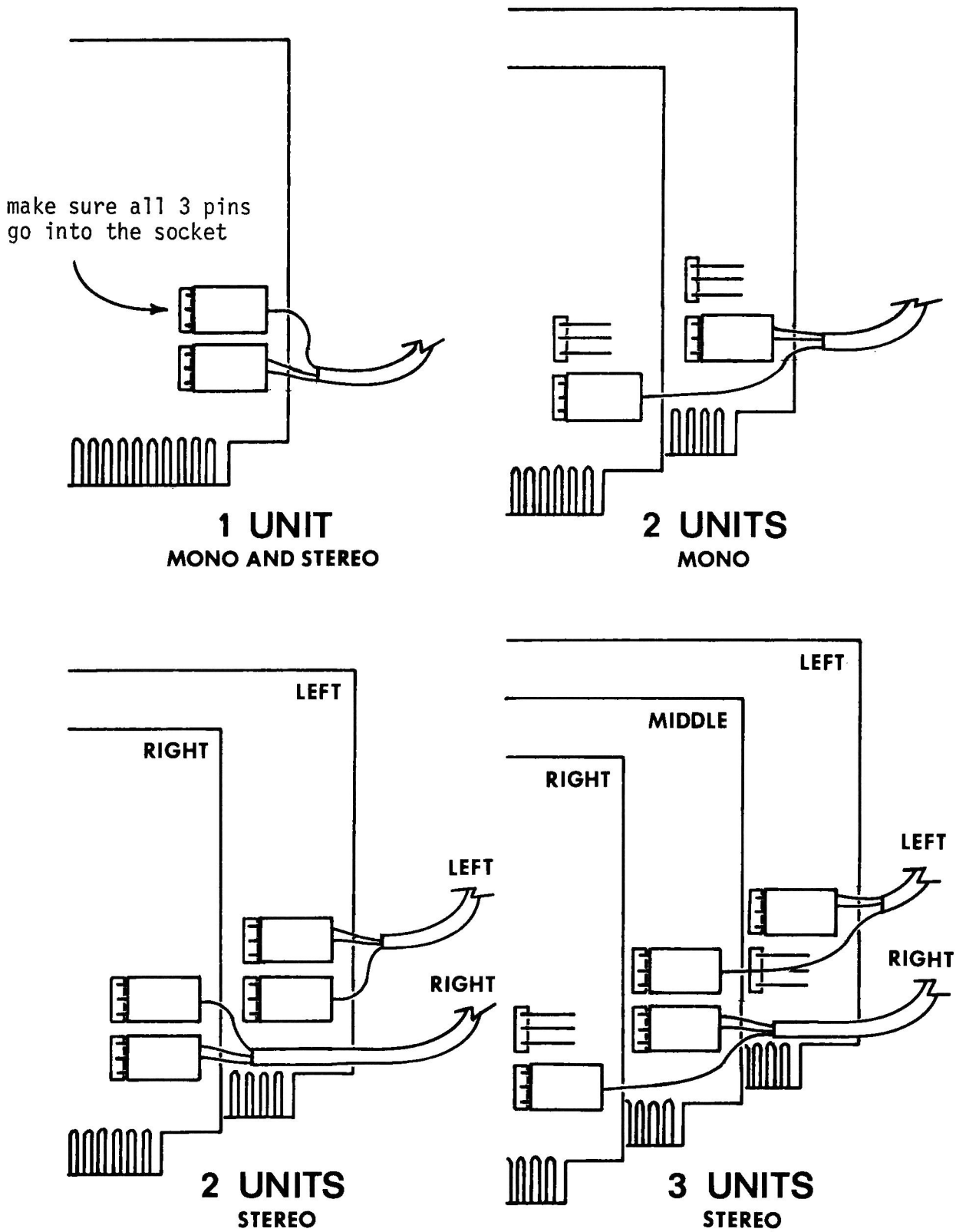
10-1 Photo
10-1 Access socket
10-1 Schematic terminals
10-2 Schematic
10-3 Repair diagram

11. MC16 TECHNICAL

11-1 Photo
11-1 Access socket
11-2 Audio outputs
11-2 Schematic terminals
11-3 Schematic
11-5 Repair diagram

INDEX

1
INSTALLATION



MC16 CABLE CONNECTION

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 card is easy. Just follow these instructions:

1. You will need an audio amplifier and speakers or a home hi-fi system. The MC1 requires a stereo amplifier (a Y adapter or a special cable, order number 1Ø-1-2, can be used for connection to a monophonic amplifier). One to three MC16 cards can be used with a stereo amplifier, and one or two MC16 cards can be used with a monophonic amplifier; see the diagram on the opposite page. Turn your amplifier off and the volume all the way down.

2. Turn the Apple off and remove the top cover.

3. Route the audio output cable(s) through one of the holes in the back of the Apple. Attach the cable(s) to the music card(s). You'll notice that the connector(s) on the end of the audio cable can be plugged into the 3-prong connector(s) on the music card(s) in either of two ways: with the slots in the plastic housing toward the circuit card or away from it. You may plug them in either way. Just be sure all three prongs go into the three holes in the plastic connector. On the MC1, reversing the connector will reverse the stereo output (Left becomes Right and vice-versa.)

4. Plug the music card(s) into expansion slot(s). Any slots may be used, but when using more than one MC16 card they must be plugged into adjacent slots. Replace the top cover of the Apple.

5. Plug the audio cable(s) 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). (There will be only one plug to plug in when using a monophonic amplifier.) When using the music card, set the amplifier to select the input used (Aux or Tuner, or Tape for "tape play" or "tape in").

6. The music card is supplied with several programs on disk. (Most programs can be loaded from disk and saved on cassette tape for use on systems without a disk.) Each program which uses the music card has a line which contains

information regarding the slot number of the card. This line is always located at line 1Ø. The first time you boot the software disk, the start-up program will ask you which slot(s) your music card(s) are in and modify line 1Ø in all appropriate programs. **IMPORTANT:** you must not stop this configuration program while it is modifying the programs or some programs may be lost. Wait for the menu to reappear. If you ever change which slot(s) your music card(s) are in, select the "reconfigure programs" option from the menu.

If for some reason you wish to change line 1Ø in a program yourself, you must load the program, change line 1Ø, and then save it. The exact procedure required for each program is explained in that program's section. If you change 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. Some MC16 programs also have a "units" setting in line 1Ø which indicates the number of MC16 cards being used. This is also set by the boot-up program. If you wish to change the line yourself, the following SLOT and UNITS values should be used:

		UNITS=1	UNITS=2	UNITS=3
SLOT=Ø	MC16's in slots:	Ø	Ø, 1	Ø, 1, 2
SLOT=1	MC16's in slots:	1	1, 2	1, 2, 3
SLOT=2	MC16's in slots:	2	2, 3	2, 3, 4
SLOT=3	MC16's in slots:	3	3, 4	3, 4, 5
SLOT=4	MC16's in slots:	4	4, 5	4, 5, 6
SLOT=5	MC16's in slots:	5	5, 6	5, 6, 7
SLOT=6	MC16's in slots:	6	6, 7	
SLOT=7	MC16 in slot:	7		

7. Turn your amplifier on. You are now ready to use your music card. Adjust the volume on your amplifier once you begin playing songs. If you have an MC16 and game paddles, you may wish to begin by running the program INTRODUCTION.

8. Decide which ENTRY program you wish to use. If you are not using an Apple IIe or an Apple III, you must use the "ENTRY" program. If you are using an Apple III, you must use the "ENTRY2" program. If you're using an Apple IIe, you should use the "ENTRY" program if you have game paddles, or the "ENTRY2" program if you don't. Note that if you don't have paddles (or if you're using an Apple III), you cannot use the INTRODUCTION or ENVELOPE programs. Also, the MC16 cannot be used with an Apple III.

OPERATING TIPS

Never connect this product to any amplifier which may have a hot (electrically charged) chassis or case. Severe damage would occur to the computer and the amplifier. 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 music card from the Apple, always unplug the audio cable from the amplifier first. Similarly, plug the music card 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 card can be damaged by excessive static. The MC1 and MC16 circuit cards have been carefully designed to minimize the possibility of damage. 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 card 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 music card ever need repair, return the entire unit (including the audio cable and software) to ALF. Be sure to include a complete description of the problem. 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. Use the "reconfigure programs" option from the HELLO (bootup) menu to be sure all programs are properly configured.

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.

4. If the disk will not boot, make sure your computer is set up to boot the

correct operating system.

5. When using ENTRY, ENVELOPE, or INTRODUCTION, make sure your paddles are plugged in and working properly. (ENTRY2 does not require paddles.)

6. If you're using an Apple IIe, Apple III, or other computer with lower case letters, be sure CAPS LOCK is on.

COPYRIGHT LAWS

Federal copyright law permits the owner of this product to make backup copies of each disk supplied with the music card. Giving such a copy away or selling copies is a crime under Federal and state laws. Modifying the programs for use with other music cards or for any purpose other than adjusting the programs to run on your computer is also illegal. Backup copies must bear a copyright notice matching that on the original disk. (A circled C or the word "copyright" indicates a copyright on programs or other text, and a circled P indicates a copyright on music recordings.) Circle-P copyrights also cover the audio output of the music cards while a copyrighted song is being played.

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

2
ENTRY

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

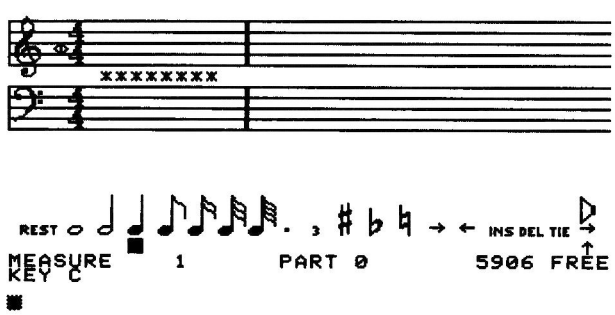
To run this program, you must have 24K or more memory. If you are using a DISK II, you need 32K or more. Very detailed graphics are presented on the screen, so it is recommended that a black and white monitor (such as the Sanyo VM42Ø9 or VM4215) be used rather than a television set, although good results 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.

Normally, ENTRY will be set for the correct slot by the boot-up configuration program. However, if you wish to change which slot the music card is in, you may change line 1Ø yourself. Begin by loading the program, then list line 1Ø. It will be 1Ø SLOT=4 for an MC1 or 1Ø SLOT=4 : UNITS=1 for an MC16 (the numbers may be different.) Carefully retype the line changing only the appropriate slot/units digit(s) to the proper digit(s) for your system. Now save the program on your disk (or cassette tape).

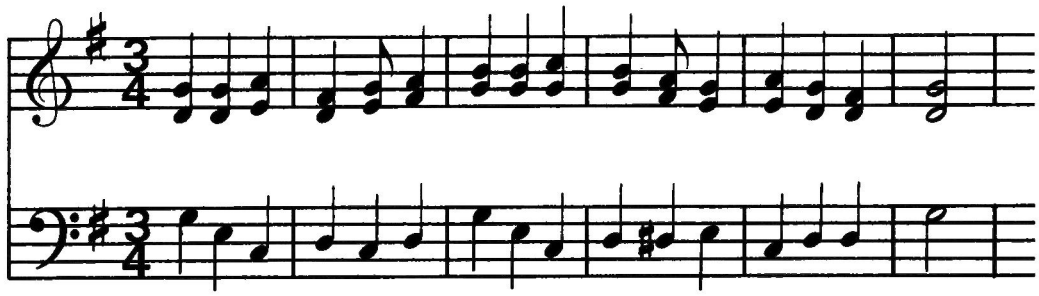
ENTERING A SIMPLE SONG

To run ENTRY, type RUN ENTRY and press return (or, from cassette tape, load the program and type RUN and press return). The screen will go to hi-res graphics mode and display:

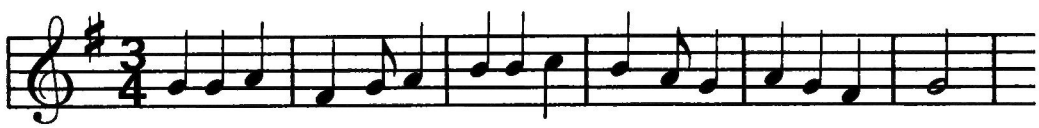


The number in front of "FREE" will vary according to memory size and other factors. It indicates the number of notes which can be added, and will be constantly updated as you enter and edit the song.

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.

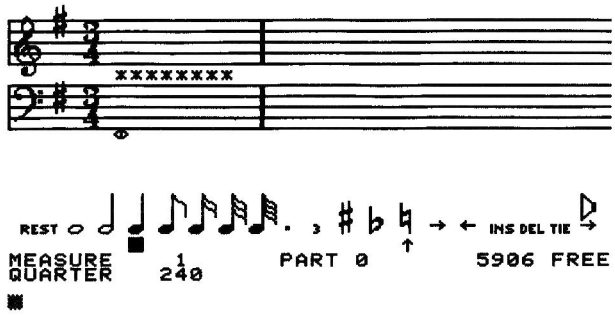


To begin entering a new song, type NEW and press return. ENTRY will display "NUMBER OF PARTS?". Just press return. This will make the song have only 1 part (part Ø). ENTRY now displays "INITIAL SPEED?". Since we don't really know what the playback speed should be yet, just press return. ENTRY will assume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

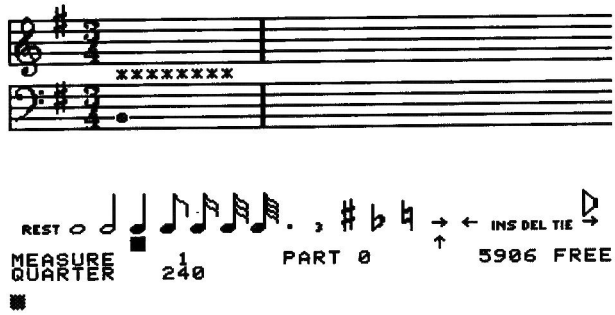


You've only been at this for a few seconds, and already you've told ENTRY two very important facts about "America", the song you're entering. Without these details it would be very difficult to enter the song properly. Why, you're probably half way to being a professional musician, if you weren't one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

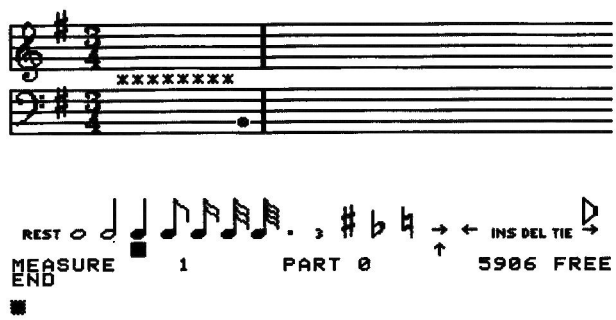
Now the cursor is at the first of eight asterisks (*) displayed between the treble and bass staves, and the item the cursor is at is a QUARTER 24Ø. These eight items are special goodies that describe things about the song which don't display well in sheet music format. This particular one indicates how 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.

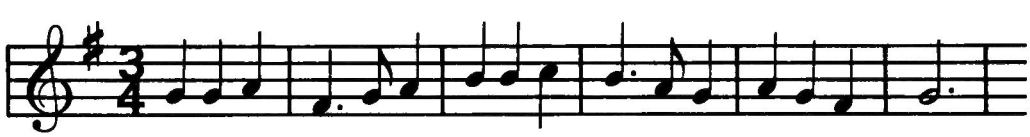
For example, using your left hand only, position the menu paddle so that the upward pointing selection arrow points to the right pointing arrow in the menu. The screen will look like this:



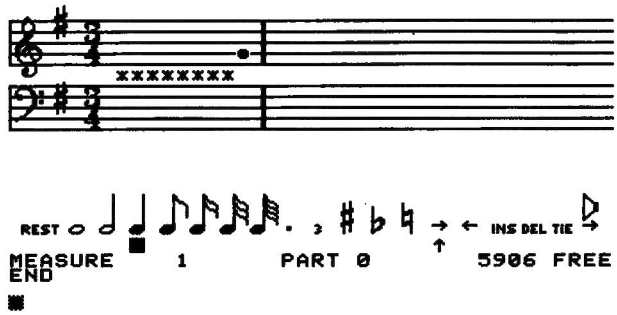
(The position of the flying saucer and the number of notes of FREE space available may be different than shown.) The right pointing arrow is used to move the cursor to the right. To cause a right movement, press the menu button using your left hand. The cursor will move right from an asterisk meaning QUARTER 24Ø to an asterisk meaning GAP 2Ø (GAP 65535 on the MC16). 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 for the MC1 or: ATTACK 8192, DECAY 5Ø, VOLUME 55ØØØ, SUSTAIN Ø, and RELEASE 5Ø for the MC16. 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:



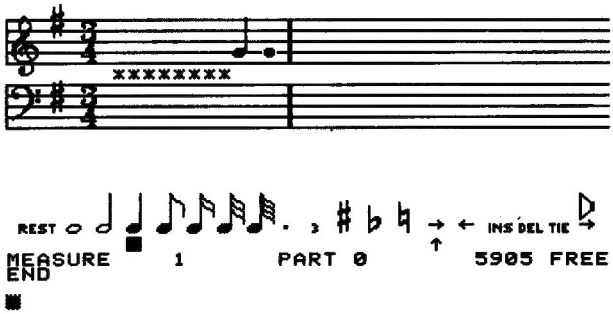
If it doesn't, you probably didn't start with RUN ENTRY like you should have. (The position of the flying saucer and the upward arrow, and the FREE number are not important.) Ready to really get into entering sheet music? Here's part Ø again, just as a reminder:



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 music card plugged in and your amplifier set up right. The screen now looks like this:



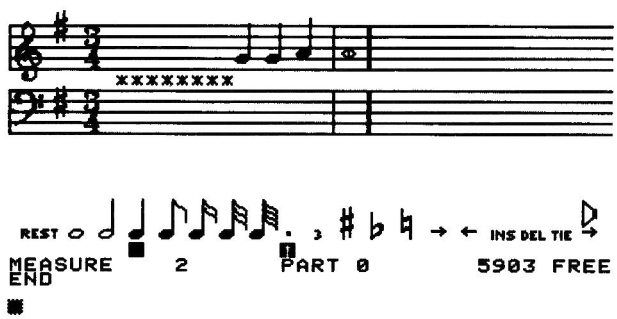
Normally when you type in something like TIME:3/4 or when you press the note button, the time signature (or note or whatever the cursor is pointing at) is written over and thus erased. However, erasing the end marker is not fun, so ENTRY automatically inserts the note (or whatever is entered) in front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

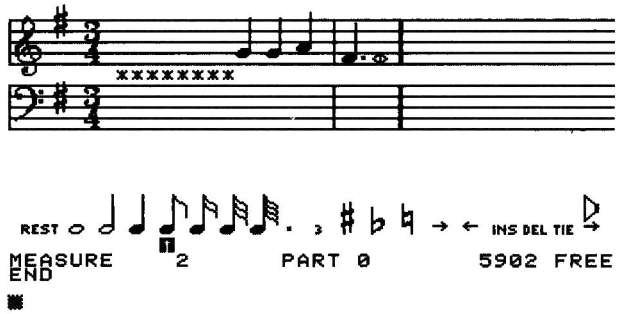
flying saucer is on the second treble line. (If not, move it until it is.) Press the note button. The next note is heard and appears on the screen. It is the same as the first note. Now, turn the note paddle until the saucer moves up one click to the space above the second line. Press the button to enter 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:

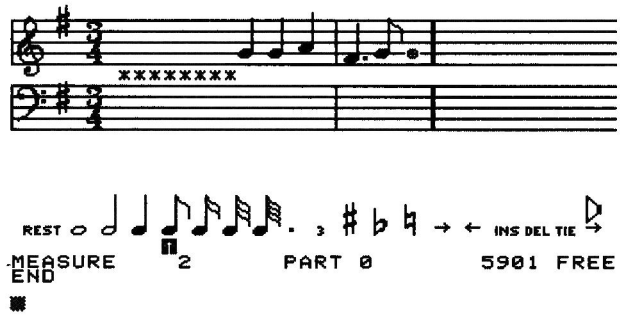


Okay, fire away. Move the note paddle down two clicks to the space under the second treble line, and press the note button. You see how you switch between the left and right hand, usually rotating a knob and pressing a button with the same hand? Since you generally keep your hands on the two knobs, you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

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:

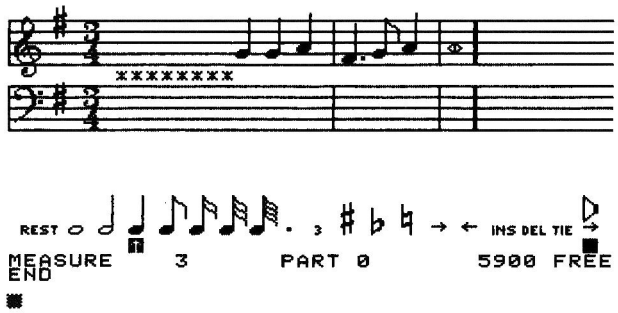


Let's take a look back. Move the cursor left one. (You know how to do it, we just did it a while back to see the first note displayed as NOTE GN3 24Ø.) The eighth note shows up as NOTE GN3 12Ø. It's the same as the first note in this part except it's half as long (only 12Ø time periods). That dotted 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 playback. You don't believe me, do you? All right, type PLAY:P and press return. ENTRY shows "SET SPEED (255) AND PRESS BUTTON". Crank the menu

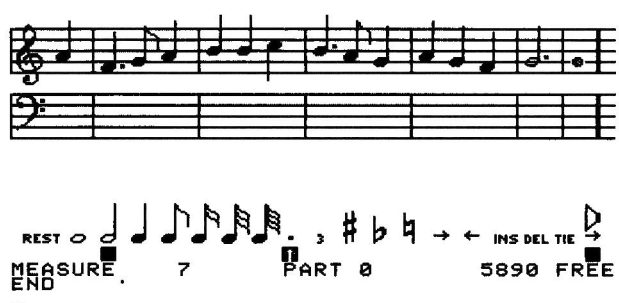
paddle up all the way (it may not actually get up to 255, but who cares?). ENTRY doesn't happen to mention which button you should press, but it is the menu button. Trust me. Punch it and ENTRY will play the song. A little slow, perhaps, but we'll know better next time.

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:

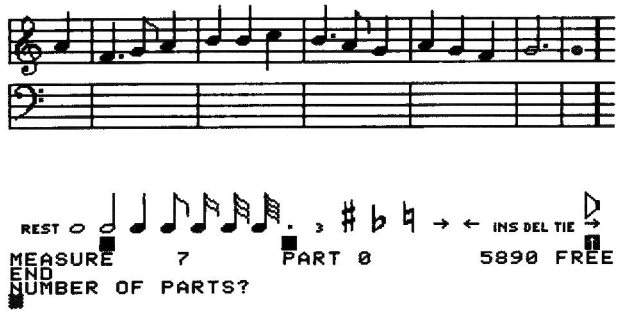


Type PLAY:P and press return. Let's try a speed of about 2ØØ now. Adjust the menu paddle to some number in the vicinity of 2ØØ. (Don't get too picky, it's not important to get exactly 2ØØ.) Punch the button, and the first 6 glorious

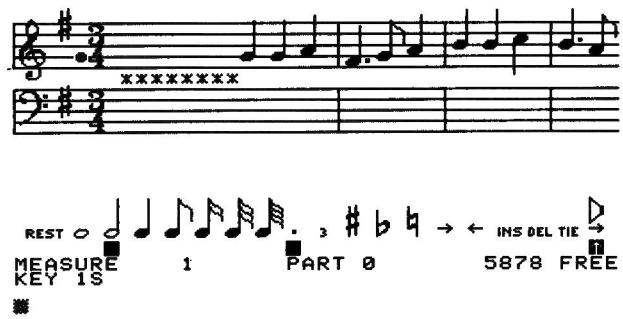
measures issue forth.

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!

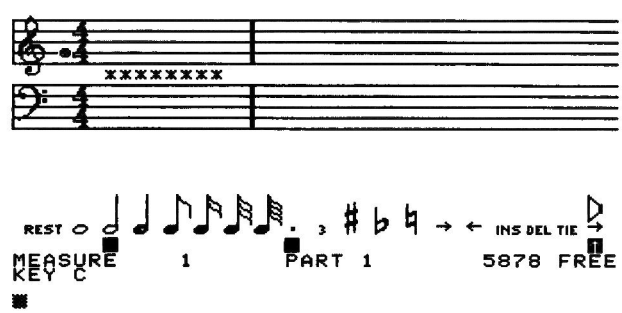
Fortunately, it is quick. First, we have to tell ENTRY that we want to add a second part. Type EDIT and press return. ENTRY responds by showing:



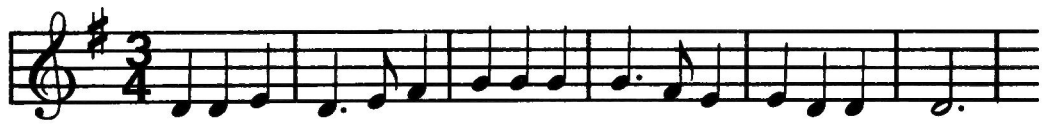
Since we want 2 parts, type 2 and press return. ENTRY then asks for the "initial speed". We've found the speed should be 2ØØ, so type 2ØØ and press return. ENTRY will then display each of the four title lines. Just press return each time. The screen now shows:



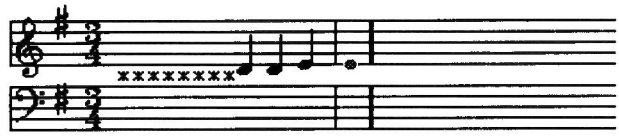
This is the beginning of Part Ø, the part you just entered. The part just created is Part 1. To see Part 1, type PART:1 and press return. The screen shows:



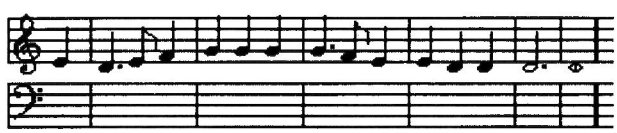
This is just like Part Ø looked originally, except there are fewer notes of "free" memory, and the screen shows "PART 1" instead of "PART Ø". You now proceed in the same fashion as before. Type KEY:1S (return) and TIME:3/4 (return). The music for Part 1 is as shown below:



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:

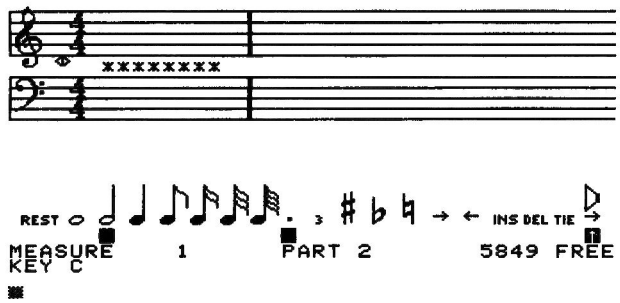


Type PLAY and press return. (Now that we've set the appropriate "initial speed", we can type PLAY for automatic playback rather than PLAY:P for paddle-controlled tempo.) You'll notice that only the first measure is played. Playback always stops when the end of the highest numbered part is reached. Since 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:



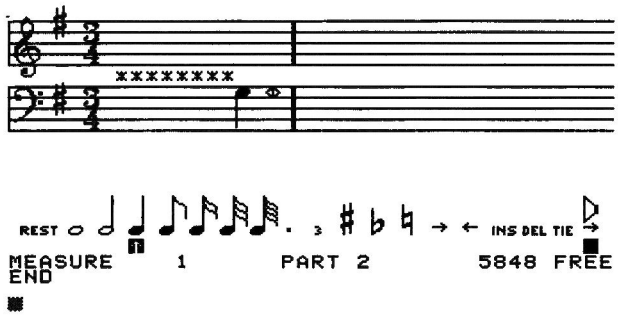
Type PLAY and press return. If there are any wrong notes, back up and correct them. (More details on correcting wrong notes will be given later in this section.) You're now ready to enter the third part.

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:

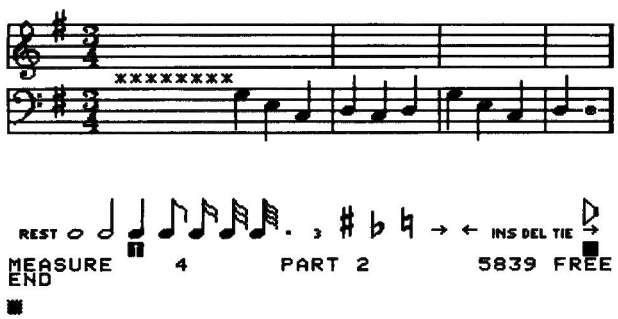


Begin as usual, typing KEY:1S and TIME:3/4, then skip the asterisks. Just for fun, type PLAY and press return. There is a brief flash, then the hi-res graphics screen reappears. This is because the end of the highest numbered part (now Part 2) is reached immediately, since there are no notes entered 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.

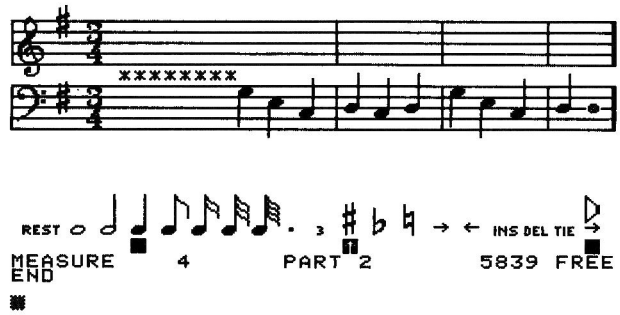
Enter the first note. The screen now shows:



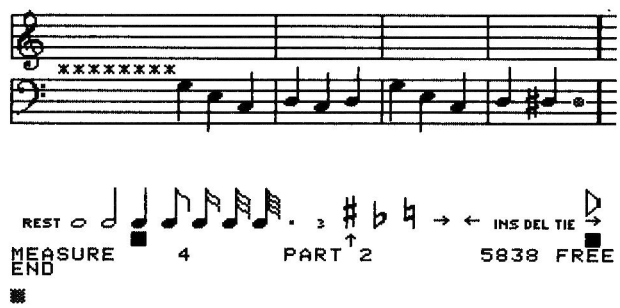
Enter the next nine notes. The screen shows:



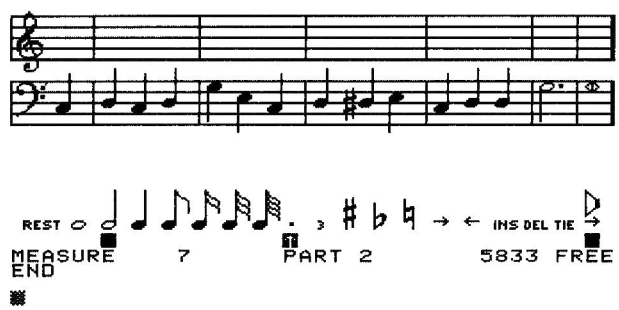
The next note is sharp, so use the menu paddle to light up the sharp sign in the menu, like this:



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



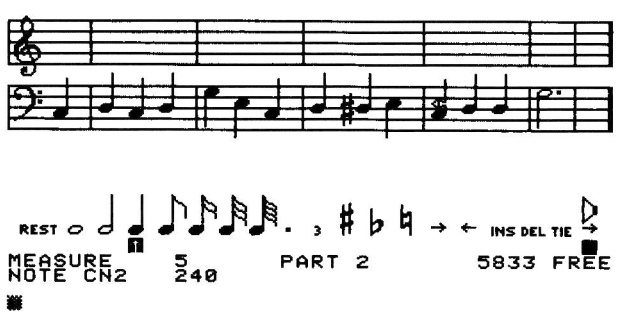
Enter the rest of the part. The screen shows:



Type PLAY to hear the song and check for errors.

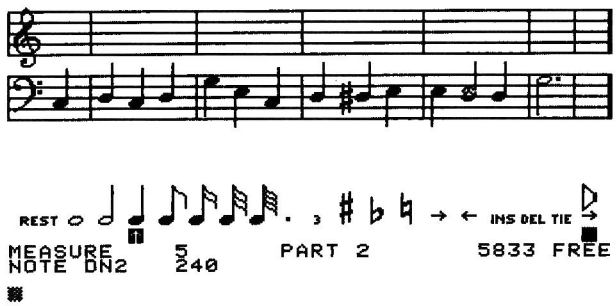
CORRECTING MISTAKES

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:

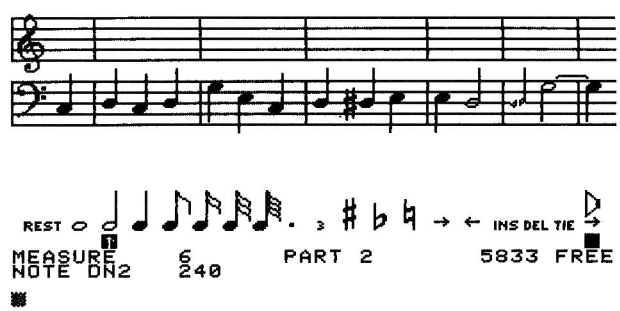


Now just press the note entry paddle button (paddle 1, of course). The old note

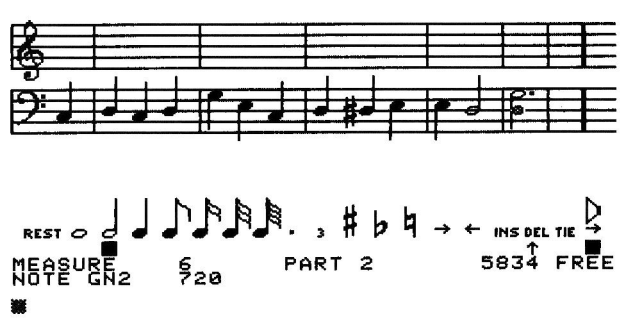
is written over by the new note:



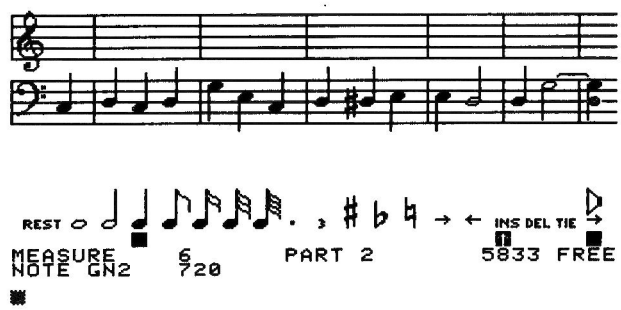
The rest of the song is not affected. Now, let's say we want to change the next note in the measure into a half note of the same pitch. Set for half note, position the cursor so it is over the quarter note's head (in order to get the same pitch), and press the button:



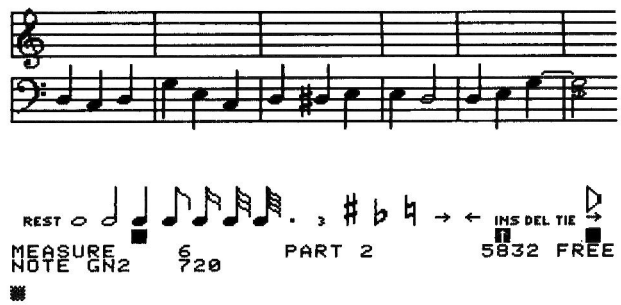
What if we want to get rid of the first note in measure 6 (where the cursor is now)? Just position the arrow for "DEL" and press the menu paddle button:



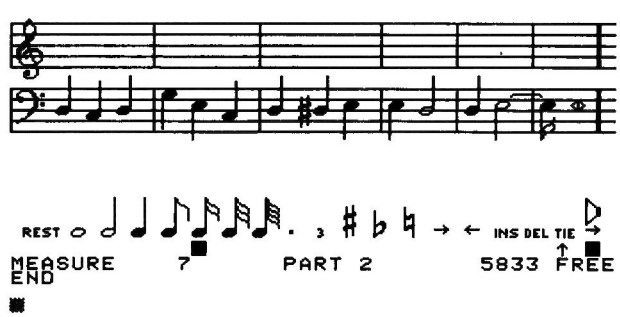
Now, let's change our mind and put it back. It was a quarter note, so set for quarter. Position the cursor on the middle bass staff line to get the same pitch. We need to insert the note, so put the menu arrow under "INS" and press the menu button to light up a block under it. Now just press the note 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:



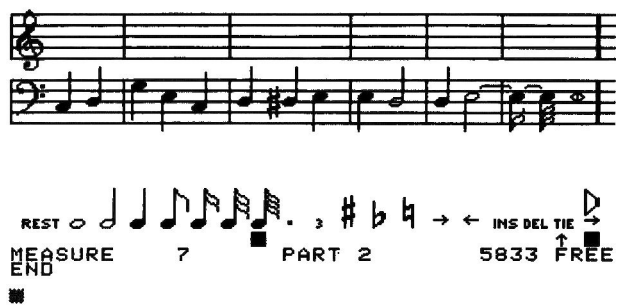
Click the note paddle up one, and press the note button again. Another note is thus inserted:



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:



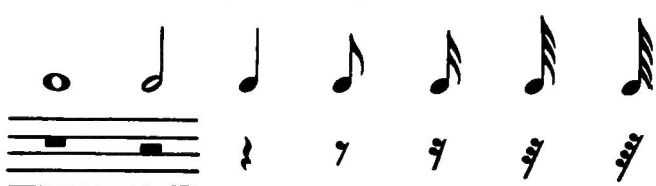
The vertical position of the note paddle cursor is not important during a "tie" since the note paddle is not used. It is important to note that although the half note tied to a sixteenth note is shown as "two" notes, it is really only one. If you back up and look at it, you will see that the length shown is 54Ø time periods, which is a half (48Ø) plus a sixteenth (6Ø). 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.

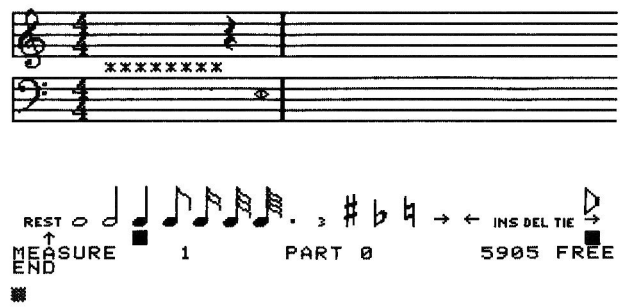
ENTERING RESTS

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:

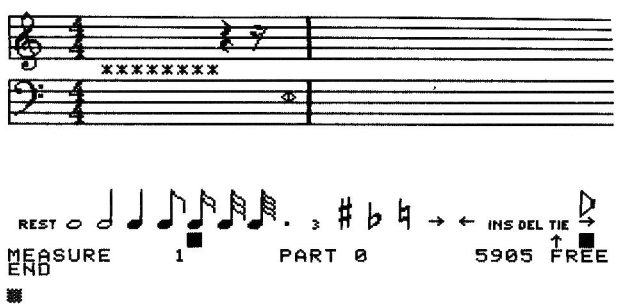


Let's start on a new song. (Actually, "song" refers to a musical composition with lyrics. Technically, one shouldn't use "song" to refer to just any melody, but

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.



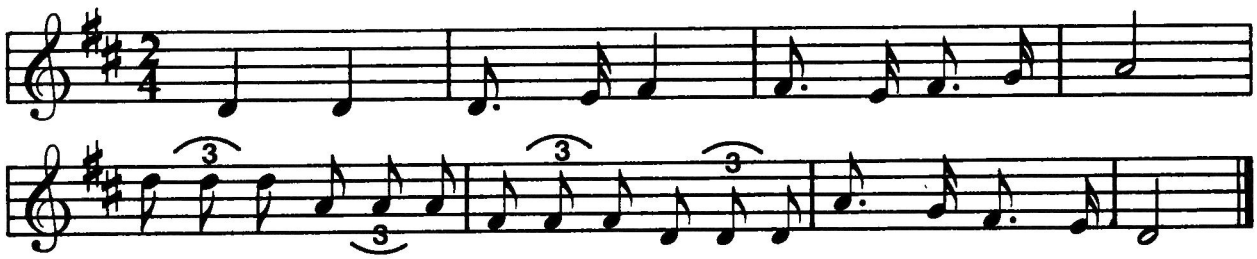
Now select sixteenth note duration and tie it onto the quarter rest. Oddly enough, the screen shows:



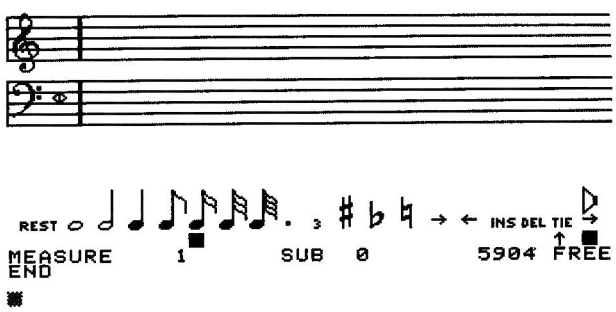
In traditional music notation, rests are never shown as being tied. This is because there is no difference between, for example, a half rest and two quarter rests during performance. The ENTRY screen display makes no distinction between a rest which is as long as a quarter plus a sixteenth, and two rests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

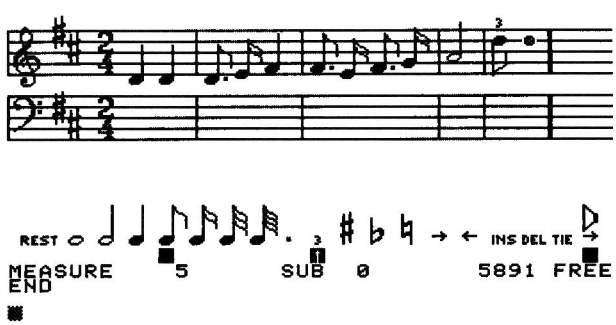
Most people are familiar with the song "Row, Row, Row your Boat". If you're not, become so. This song plays the same theme several times, and from several parts. It seems that one would have to enter this theme several times. Since repeated sections such as this are common in music, ENTRY has special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



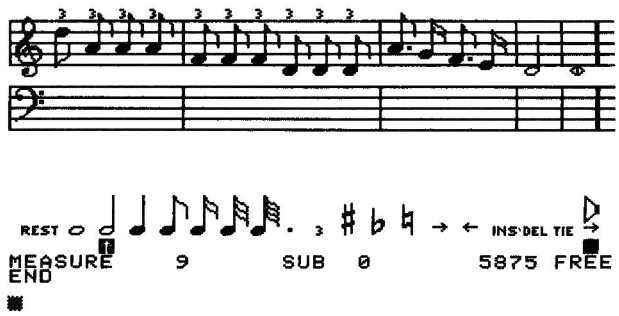
This theme must be entered in a special fashion which allows it to be played many times. This is done using a subroutine. Type NEW and press return several times (as usual) to start fresh. Now type SUBROUTINE:Ø and press return. The screen will show:



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.

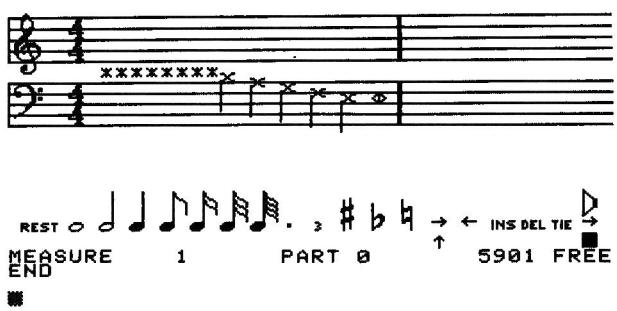
The music card is supplied with a few sample songs which can be loaded and played. Additional songs are available at extra cost.

ADJUSTING THE TEMPO

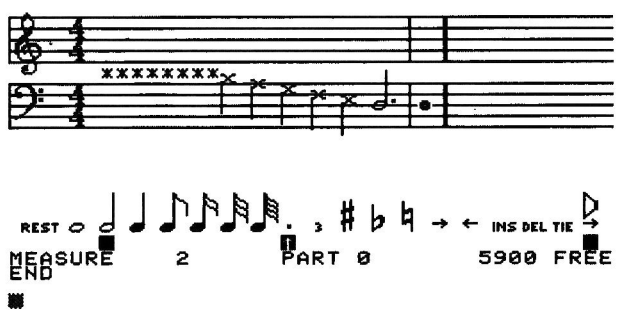
Let's say we want to enter the "row" theme to play twice as fast with the same initial speed setting ("initial speed" is changed to make smaller adjustments). That means each note will have to play for half as many time periods. Type NEW and press return as required, enter the key and time signatures, 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. 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 <u>insert</u> 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 <u>new</u> 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 music card 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 music card 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 (256 on the MC16) is the actual volume setting on the music card 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.

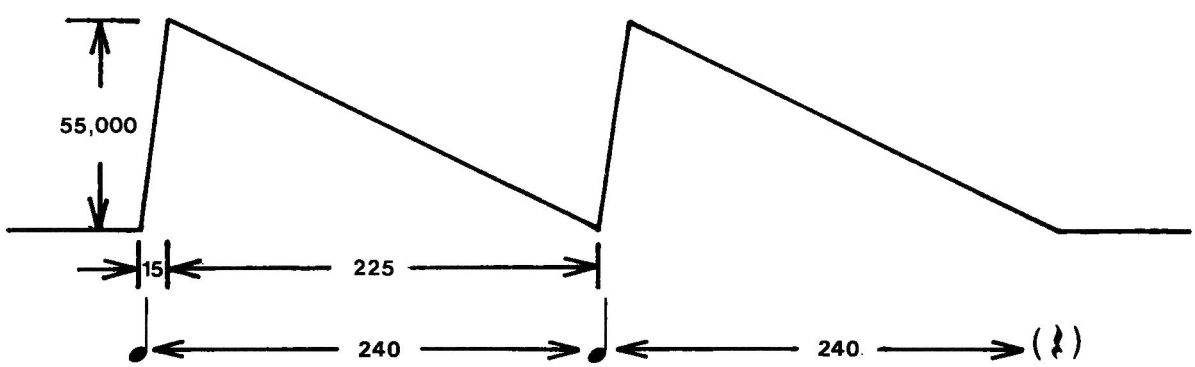
Notetrinos generated using a high-power paramatron at the University of Northern South Dakota (just across the border from Hoople) have revealed the following characteristics of these settings. (What?) When a new note begins, the most recent decay setting is written into the "current decay" rate, the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

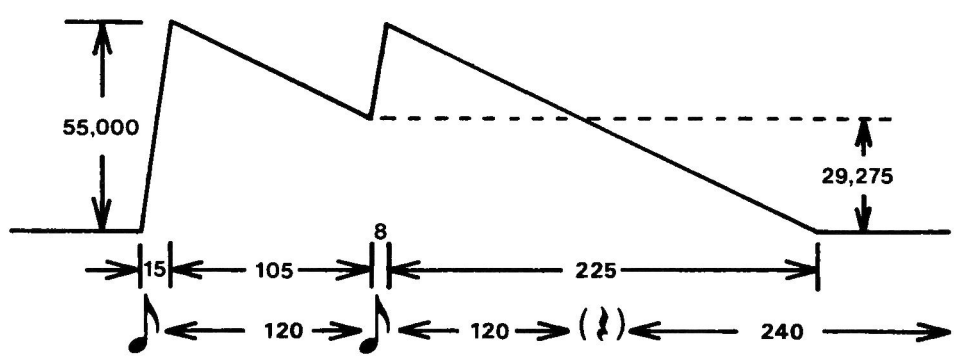
This simple process generates a variety of complex envelopes, for single notes or for several. Be ye not confused: each note does not necessarily have an

"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.

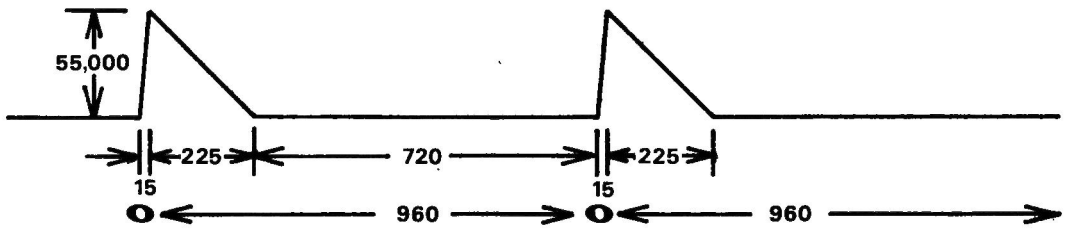
Some examples are in order. Let's say we want a simple AD (attack-decay, or "ping") envelope with a volume level of 55ØØØ. Further, let's say it is a quarter note with standard QUARTER settings (24Ø time periods) and we want the first 16th of the note to be the attack stage, and the remaining 15/16ths 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:



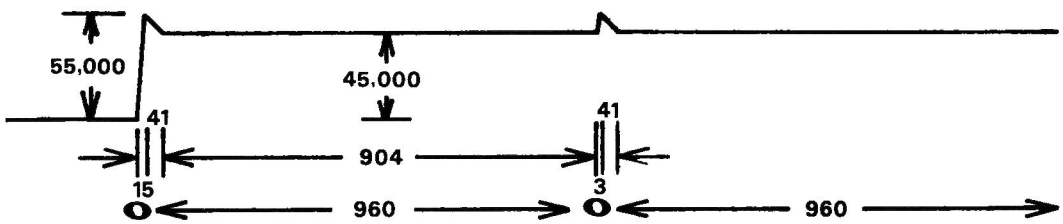
The GAP setting must be 65535 to avoid a release stage. Now, what if we played an eighth note with this setting? The loudness contour would appear thus:



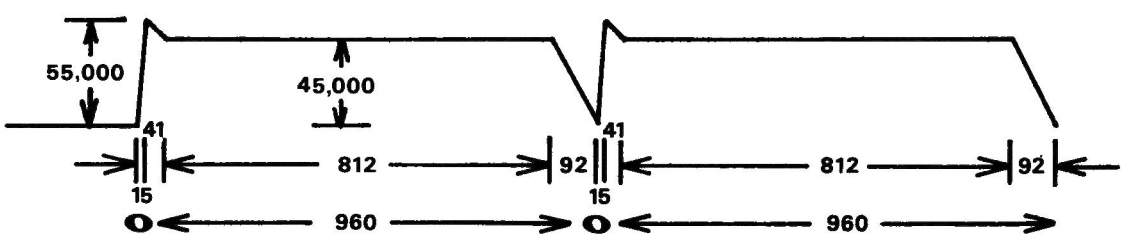
If an eighth note is followed by a rest, the release stage will begin. Therefore the release setting should be set to the same as the decay setting, unless you want something different to happen on notes followed by rests. What if we played a whole note? Behold:



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).

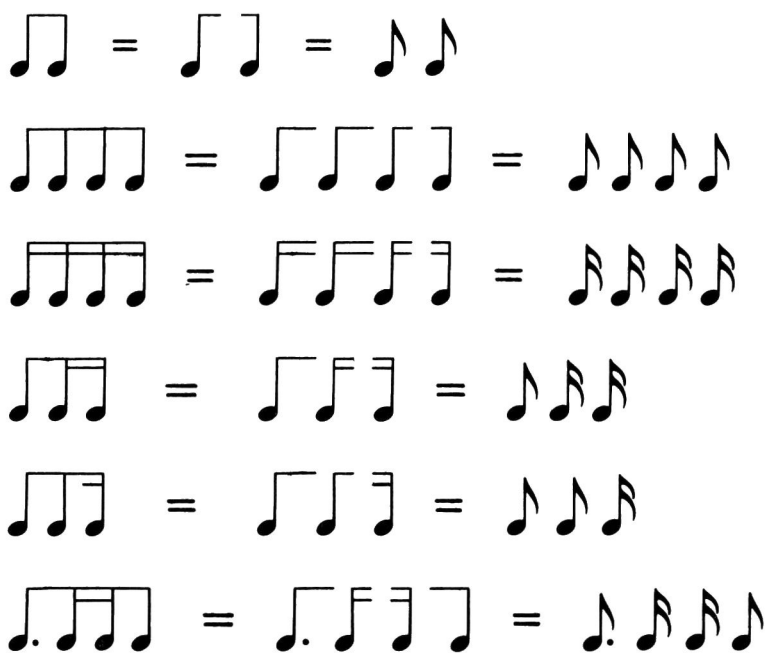
Experiment more with the settings. Draw graphs like the ones above if they help you. Look at other people's envelope settings if you run out of ideas. Here's a real tip: program what would normally be a whole part into a subroutine instead. Then you can call it from two parts, and use different envelope 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.

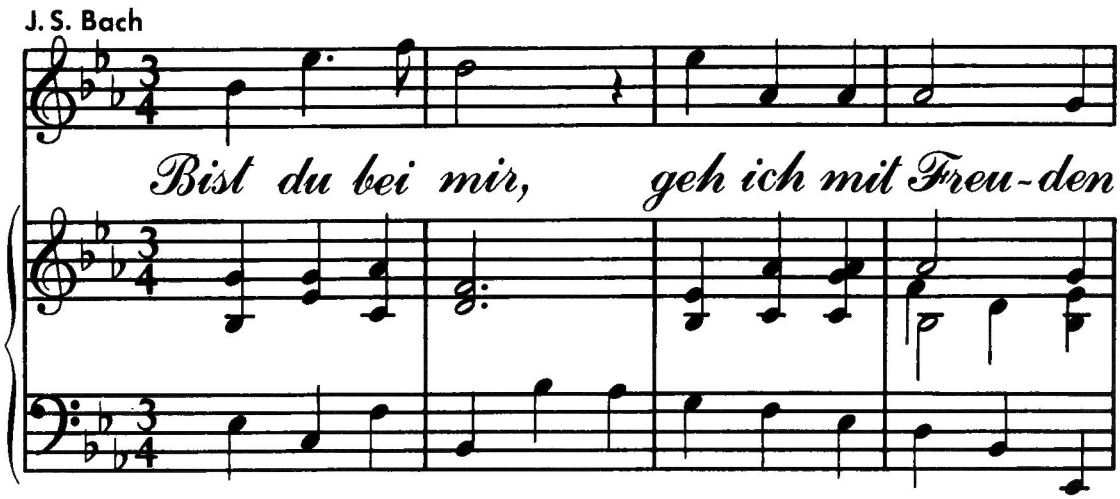
BEAMING

In conventional sheet music notation, groups of notes shorter than quarter notes are often connected together with "beams". This makes them easier to read during performance. The beamed notation can easily be translated into regular "flagged" notation as shown below. ENTRY uses the flagged notation because it requires less memory per note, and can be entered more quickly.

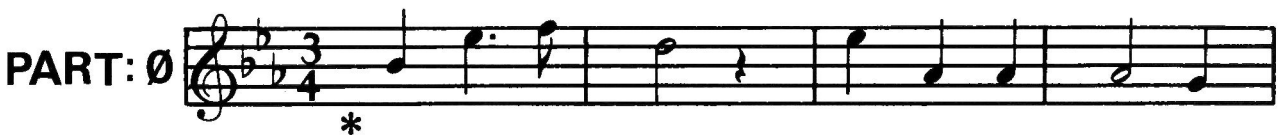
IF WRITTEN: **ENTER:**



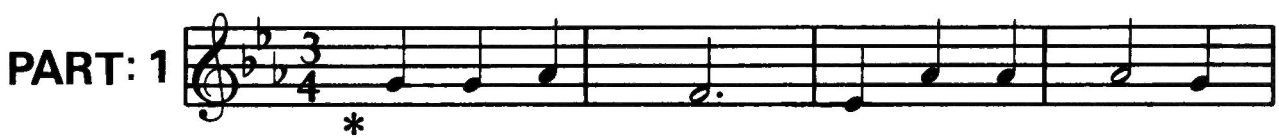
etc.

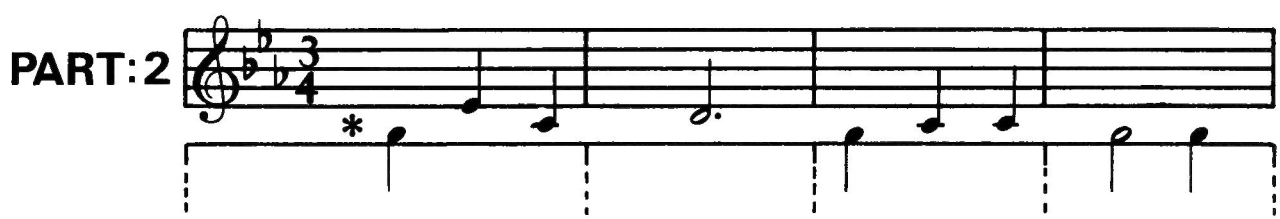


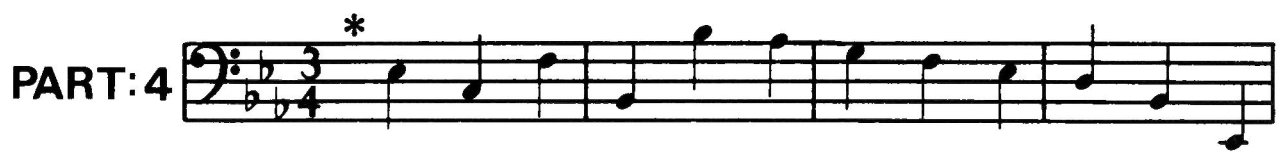
BECOMES:



(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, VOLUME can be shortened to just V since no other keyword starts with V. 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 music card.

FP

The FP command is used to exit ENTRY and return to BASIC. The current song

data is lost. Note that after FP is used to exit ENTRY, it cannot be rerun simply by typing RUN. It must be reloaded (using RUN ENTRY) to be run again. Sample command: FP (return).

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).

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

28672-32767 light blue, 32768-36863 brown, 36864-40959 orange, 40960-45055 grey, 45056-49151 pink, 49152-53247 green, 53248-57343 yellow, 57344-61439 aqua, 61440-65535 white (loudest). (Based on Apple's suggested color names; actual colors may vary.) Ignoring the fact that there are two colors named grey, each color represents one of the 16 different actual volume settings on the MC1 music card. On the MC16, each color represents a range of 16 volume settings since there are 256 total volume settings available. "PLAY" plays the song using the "initial speed" and RATE command to select playback speed. "PLAY:P" plays the song using the paddle to select playback speed (allowing continuous variation). NOTE: both PLAY commands change (a) the CHANNEL function settings and (b) the subroutine FE bytes. These changes will not be apparent to the ENTRY user, but could affect PERFORM users. See the PERFORM section for additional information. Sample command: PLAY (return).

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

*****DISK[:<comment>]**
The ***DISK command increases the karma of the user when using DOS 3.1. This command has no effect when using DOS 3.2, DOS 3.3, or a cassette based system. Sample command: ***DISK: FILE NOT FOUND ERROR (return).

TYPE 2 COMMANDS

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

DEL

The DEL symbol is used to delete the item the cursor is currently at. It is requested by pressing Paddle 0's button while the upward-pointing arrow is aimed at DEL. When requested, the item the cursor is at is deleted from the song data. If it is a note, it is sounded through the music card 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?, INITIAL 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 INITIAL SPEED must be a number from 1 to 255. 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 (on the MC16, GAP is 65535, DECAY is 5Ø, and RELEASE is 5Ø). (All subroutines and parts always end with an end marker.) Stereo is set to the standard values:

STEREO:MLRMLRM*L*R* for the MC1 or STEREO:2,LRLRLR and STEREO:3,MLRMLRMLR for the MC16. The NEW command then asks for the INITIAL SPEED? which can be given as any integer from 1 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> (MC1) **STEREO:<2-3>,<string>** (MC16)
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.

On the MC1, a star (asterisk, *) may be typed following the L, M, or R for each part 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.

On the MC16, *'s cannot be used since there is no fuzz feature. Stereo is available only when using two or three MC16's. STEREO:2,<string> sets the stereo for playback with 2 cards, and STEREO:3,<string> for playback with 3 cards. On songs having 6 or fewer parts, both the 2 and 3 settings can be specified; on songs have 7 to 9 parts only the 3 setting can be used. M's (for Middle) cannot be used in the 2 setting since there is no "middle" card when 2 cards are used. Note that if timing mode will be used, there can be only two R's instead of three

in the 2 setting or two M's in the 3 setting because the remaining R or M channel is used for timing mode.

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 for the MC1: STEREO:MLMR (return). Sample command for the MC16: STEREO:3,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 affect 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
On the MC1, the FUZZ command is used to select fuzz (white noise) mode or normal mode. FUZZ mode is not available on the MC16. 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 (Ø-191 on the MC16) 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).

RATE:<Ø-255>
The RATE command changes the playback tempo. It may be included in any part, but it changes the playback tempo for all parts. (To avoid confusion, you may want to use RATE commands in part Ø only.) The number of time periods per second is approximately 92,773/speed. However, the RATE setting is slightly different from the initial speed setting, and is computed with the formula: speed-11*(number of parts). For example, in a 3 part song with an initial speed of 2ØØ, the equivalent RATE setting would be 2ØØ-11*3, or 167. Using a RATE commnd with a value smaller than 167 would begin a faster tempo, or a RATE larger than 167 would begin a slower tempo. Also see "SPEED/RATE SETTINGS" under "TIPS". Sample command: RATE:1Ø1 (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>
(MC16 only.) TEMPO commands are no longer used. They can generally be replaced with RATE commands. The conversion formula is: RATE=TEMPO/19.17-1-11*(number of parts).

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.

SPEED/RATE SETTINGS

Speed (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 speed setting is high enough. Generally, speed settings lower than 15Ø (or equivalent RATE settings) are never used. Songs having many parts active and using several levels of subroutines may require even higher settings.

QUARTER and speed settings can be determined from the metronome settings shown on some sheet music. Usually the metronome setting is shown with a quarter note, an =, and a number; that's the number of quarter notes per minute. If the note shown is a half note, you'll need to multiply the number given by 2 to get quarter notes per minute (and so on for other possible notes). Using the chart below, select a QUARTER setting based on the quarter notes per minute number

from the sheet music:

QUARTER NOTES PER MIN.	QUARTER SETTING	SPEED COMMAND
4Ø-41	624	SPEED:13/5
42-45	576	SPEED:12/5
46-49	528	SPEED:11/5
5Ø-55	48Ø	SPEED:2
56-62	432	SPEED:9/5
63-71	384	SPEED:8/5
72-83	336	SPEED:7/5
84-99	288	SPEED:6/5
1ØØ-125	24Ø	
126-167	192	SPEED:4/5
168-2Ø8	144	SPEED:3/5

You can either pick the QUARTER setting before you start entering your song (and change the QUARTER setting at the start of each part before entering notes), or you can enter the song with QUARTER:24Ø as usual and use the SPEED command from the chart above to modify the song. CAUTION: be sure you have a QUARTER command in all subroutines, and be sure to SAVE your song before using SPEED (see the SPEED command) in case the results are not as desired. Now compute the paddle setting with the formula: SPEED=(5577465/(QUARTER*METRONOME))-1 where QUARTER is the QUARTER setting and METRONOME is the number of quarter notes per minute. This "SPEED" value can be used directly as an INITIAL SPEED or a paddle setting, but you must subtract 11 times the number of parts from the SPEED value to obtain the proper value for the RATE command.

"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 Applesoft.

MC1 RANGE
The lowest note on the MC1 is the C two octaves below middle C, the MC16 plays over an octave lower. The MC1 playback routines will transpose notes which are too low to play up by one or more octaves.

MC16 PARTS LIMIT
Each MC16 card can play three simultaneous tones, and thus is limited to three parts (PART:Ø, PART:1, and PART:2). With two MC16's, six parts can be used; and with three MC16's all nine parts are available.

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.

ENTRY2

The ENTRY2 program is a new version of ENTRY designed for use on Apple IIe computers without paddles or on Apple III computers. (The paddles circuit of the Apple III is so different than the Apple II, II+, and IIe circuits that none of the programs in this package that require paddles will function with Apple III paddles.) ENTRY2 cannot be used on Apple II, Apple II+, or similar systems.

The main difference between ENTRY2 and ENTRY is that the four arrow keys, the open Apple key, and the closed Apple key (or space bar on the Apple III) are used in place of the game paddles.

Generally, the instructions for the ENTRY program apply to the ENTRY2 program as well. Where the ENTRY instructions say to turn the paddle 1 knob (or the "note paddle"), you should use the up arrow key or down arrow key instead. Up arrow and down arrow will move the "flying saucer" cursor up and down. Where the ENTRY instructions say to turn the paddle Ø knob (or the "menu paddle"), you should use the left and right arrow keys instead. Left arrow and right arrow will move the menu selector arrow left and right except when you are typing a command. (When the typing cursor is not at the leftmost column on the screen, left and right arrow can be used for editing in the usual fashion. When the typing cursor is at the leftmost position, they will move the selector arrow.)

When the ENTRY instructions say to press the paddle 1 button (or the "note button") you should press the open Apple key instead. When the ENTRY instructions say to press the paddle Ø button (or the "menu button"), press the closed Apple key on the Apple IIe or the space bar on the Apple III. (Note: remember that space bar repeats if held down.)

References in the ENTRY instructions to use your "left hand", "right hand", or "same hand" should be ignored when using ENTRY2.

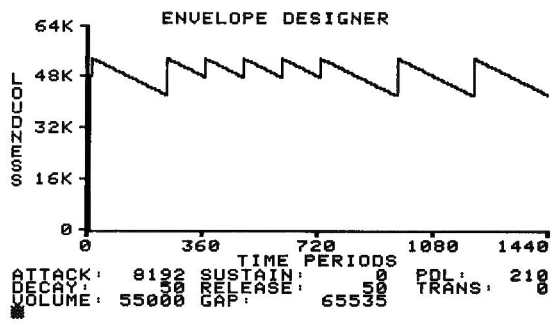
Another difference is that ENTRY2 doesn't have the "PLAY:P" command, since it is written for use without paddles. "PLAY" can still be used, of course.

If you're changing line 1Ø yourself in ENTRY2 for the MC1, you might also want to list line 2Ø. It will be either 2Ø APPLE=2 (for use with an Apple IIe) or 2Ø APPLE=3 (for use with an Apple III). This line can be changed if need be. Like line 1Ø, line 2Ø is set automatically by the bootup program or when the "reconfigure programs" option is selected (in the HELLO program).

Cassette functions should not be used with an Apple III.

3
ENVELOPE

The ENVELOPE program is used to design, hear, and view envelopes. Type RUN ENVELOPE to begin. (Game paddles are required. ENVELOPE cannot be used with an Apple III.) To help you get started, ENVELOPE has 9 pre-programmed examples (7 for the MC16). When the program is first run, it is set up with example number Ø. To hear and see this example, set the playback speed to a relatively large number (over 2ØØ) with paddle Ø, then press the paddle Ø button. A short song will be played, and the envelope (loudness contour) will be drawn on the screen simultaneously, so you can see the exact relationship between the screen and the sound. The screen will look like this:



You can play example Ø at any speed (tempo) you like just by turning the paddle Ø knob to a different setting, then pressing the paddle Ø button. The various envelope settings used to create the sound you hear are shown on the screen. By using these same settings in your song, you can create the same sound.

To hear and see one of the other examples, type EXAMPLE: and the example number (Ø to 8 for the MC1 or Ø to 6 for the MC16), then press return. After a long pause (during which the new example is set up), you can play the example in the usual manner.

One of the most useful features of ENVELOPE is that you can change any parameter and see how it affects the sound. Any of the envelope parameters displayed on the screen can be changed simply by typing the appropriate ENTRY command. For example, to change the attack rate to 1ØØØ, type ATTACK:1ØØØ and press return. (These commands can be abbreviated in the same fashion as ENTRY commands. ATTACK:1ØØØ can be abbreviated to AT:1ØØØ.) The change will be shown on the screen, and the song fragment can be played with the new settings.

ENVELOPE also has a variety of other commands. The DISPLAY command is used to change the scale of the display. By typing DISPLAY: and a number from 1 to 8, you can select a display 1 to 8 quarter notes wide. (Note: QUARTER is always assumed to be 24Ø time periods long.) Large settings show a long time duration and are useful for see the "whole picture". Small settings show a short time duration and are useful for seeing more detail.

If you prefer to type PLAY rather than press the paddle button to begin playback, you may.

Several commands relate to changing the song fragment being played. The first is the return key. Pressing just return causes the display to change from the graphic display to a text display showing the notes being played. (To go back to the graphic display, press return.) For example Ø, the text display shows:

```
                ENVELOPE DESIGNER

NOTES ARE ENTERED AS A ':'
FOLLOWED BY A LETTER FROM
A TO G.  AN S, F, OR N MAY          MELODY
THEN BE USED TO INDICATE
SHARP, FLAT, OR NATURAL.     -->1   :A3Q
THE OCTAVE NUMBER (0-7)        2   :D2E
FOLLOWS THIS.  RESTS ARE       3   :F3SE
ENTERED AS ':R'.               4   :C3SE
                               5   :C3SE
BOTH FOR NOTES AND RESTS,      6   :A3QQ
DURATION IS THEN ENTERED.      7   :D2QQ
A DURATION CAN BE ANY OF       8   :D2Q
W, H, Q, E, S, T, OR X
OPTIONALLY FOLLOWED BY
A '.' OR '3'.

ATTACK:  8192 SUSTAIN:     0  PDL:  210
DECAY:    50 RELEASE:     50  TRANS:   0
VOLUME: 55000 GAP:     65535
```

Song fragments are limited to a maximum of 8 notes, numbered 1 through 8. An arrow (-->) points to the note which can currently be changed. Two types of changes can be made. You can either type END to erase the note and all following notes, or you can simply change the note to a different note. (Note: the first note cannot be erased. "Note" means either a note or a rest.) To change a note, type : and then a note specification. A note specification is a letter from A to G which indicates the note within an octave; optionally followed by S (for a sharp note), F (for a flat note), or N (for no apparent reason); then a digit from Ø to 7 which indicates the octave (C3 is middle C); then a duration specification. (Rests are specified by the letter R followed by a duration specification.) A duration specification is W (for whole), H (for half), Q (for quarter), E (for eighth), S (for sixteenth), T (for thirty-second), or X (for sixty-fourth); optionally followed by a period (for dotted) or a three (for triplet). (As in ENTRY, dotted sixth-fourth notes are not allowed.)

For example, typing :E3Q and pressing return would change the note pointed to by the --> arrow to a quarter note whose pitch is the E above middle C. Typing :RS3 and pressing return would change the pointed note to a sixteenth triplet rest. When a note is changed, the pointer advances to the next note so it can be changed if desired. (Note: when the 8th note is entered, the pointer remains at note 8 since no additional notes can be entered.)

To move the --> pointer to any desired position, type EDIT: and the position number. For example, EDIT:1 moves the pointer to the first note.

When you are finished using ENVELOPE, type FP to exit the program.

COMMANDS

:R<duration> or :<note><duration>
ATTACK:<value>
DECAY:<value>
DISPLAY:<1-8>
EDIT:<1-8>
END
EXAMPLE:<Ø-8> (MC1) **EXAMPLE:**<Ø-6> (MC16)
FP
FUZZ:ON or **FUZZ:OFF** (MC1 only)
GAP:<value>
INT
PLAY
RELEASE:<value>
SUSTAIN:<value>
TRANSPOSE:<Ø-255>
VOLUME:<value>

<duration> is a duration letter (W, H, Q, E, S, T, or X), or a duration letter followed by a period, or a duration letter followed by a 3. <note> is a note letter (A, B, C, D, E, F, or G) followed by an octave digit (Ø, 1, 2, 3, 4, 5, 6, or 7), or a note letter followed by an accidental letter (S, F, or N) followed by an octave digit. <value> is an integer from Ø to 65535 optionally followed by a slash (/) and an integer from 1 to 65535 (see the ENTRY section).

LINE 10

Line 1Ø contains the variables SLOT, UNITS, and CARD. These are set by the HELLO program automatically. If you wish to change this line yourself, LOAD ENVELOPE, list and change line 1Ø, then SAVE ENVELOPE. If you have an MC1, use UNITS=1 and CARD=1. If you have an MC16, use the appropriate UNITS setting and CARD=16.

4
PROCESS/MLIST

PROCESS

The PROCESS program is used to do advanced editing of ENTRY-created songs. Mainly, PROCESS allows you to move large sections of music in order to change parts into subroutines, or work on a musical section as an independent song and then append it onto the main song. You begin using PROCESS by typing RUN PROCESS.

An important feature of PROCESS is that it works with two songs in memory at once. One song is called the main song. All editing actually takes place on the main song. The second song is called the auxiliary song, which is only used in certain applications. Sections can be read from the auxiliary song and appended to the main song. It is important to note that the main song and the auxiliary song can actually be the same song. This is done when you wish to read a section from one place in a song and append it to another place.

Generally you start with a LOAD command. Typing LOAD: and a song name (then pressing return, as usual) causes the specified song to be read from disk and used as the main song. (Note: if you already had a main song or a main song and an auxiliary song, it or they will be lost.) You may then use the various editing commands or the status printing commands. If you make any changes in the song, you must use the save command if you wish to have a copy of the song with these changes saved on disk. This is done by typing SAVE: and a song name. You don't have to save the song with the same name. in fact, it is usually best to save the song with a different name if you have enough space on your disk; that way you'll have a copy of the song before any changes (with the original name used in the LOAD command) and a copy with the changes (with the new name used in the SAVE command). Then if you later discover you didn't make the changes you intended to make, you can get the original back and do whatever you wish.

Only the APPEND command requires an auxiliary song. The following commands require only the main song.

The **DELETE** command is used to delete a part or a subroutine. You just type DELETE:PART and the part number, or DELETE:SUB and the subroutine number, then press return. For example, you type DELETE:SUB 5 to delete subroutine 5. (Remember that subroutine numbers always start with Ø and go up by 1's. In ENTRY you can assign any numbers you like to subroutines, but when you save the song they are automatically changed to start with Ø and not skip any numbers.) It is very important to remember that from the time you LOAD the main song to the time you SAVE it, all part and subroutine numbers will stay the same. For example, if you delete subroutine 3, subroutines 4 throught 7 (or however many

subroutines you have) do not suddenly become subroutines 3 through 6. Subroutine 4 will remain subroutine 4 throughout the session. When you SAVE the song, subroutines 4 through 7 would become 3 through 6 unless you've "filled in" the missing subroutine 3 with an APPEND command. The various calls to subroutines 4 through 7 will be changed so they still call the proper subroutines, of course. Likewise, if you delete part 2, parts 3 on will remain 3 on until the SAVE; at which time they will be renumbered so there are no missing numbers.

If you attempt to delete a subroutine that is called by some remaining part or subroutine, the warning "THIS WILL CREATE UNDEFINED REFERENCES" is printed, and you will be given a chance to change your mind.

CAUTION: if you delete a part, the stereo settings will be reset to standard settings.

The **CHANGE** command is used to change all CALL's in a specified song subset so they call a different subroutine. For example, to change all CALL:2's in all parts and subroutines into CALL:5's, you type CHANGE:2 TO 5. If you wish to change only the CALL's which occur in parts (and not those in subroutines), add IN ALL PARTS to the command. To change only the CALL's in a particular part, add IN PARTand the part number. Similarly, CALL's in a subroutine can be changed by adding IN SUB and the number. For example, to change all CALL:4's in subroutine 3 to CALL:7's, type CHANGE:4 TO 7 IN SUB 3. (Or, if you don't like to type, you can abbreviate that command to just C:4T7IS3.) If you change the CALL's to a subroutine number that doesn't exist, you will get the "THIS WILL CREATE UNDEFINED REFERENCES" message and a chance to change your mind.

The **STATUS** command prints the value of various parameters at the end of the selected part or subroutine. You type STATUS:ALL PARTS, or STATUS:PART and a part number, or STATUS:SUB and a subroutine number. For example, when you type STATUS:PART 2 the status command will look through all commands in part 2, remembering the most recent setting for each parameter and compiling similar information. The last setting for KEY, TIME, QUARTER, GAP, TRANSPOSE, ATTACK, DECAY, VOLUME, SUSTAIN, RELEASE, FUZZ, TEMPO, and the various possible POKE's will be printed. Then a list of all subroutines called, and the total time duration in time periods will be printed. Note that if any parameter doesn't occur at all in the subset scanned, it will not be printed. An example use for this command would be if you've entered half of a song, and wish to put the other half in a separate song for now so you don't have to play through the first half (which you've already perfected) to hear the second half (which you're working on). First, you do a STATUS:ALL to get a list of final parameters for each part. If you have a printer, you can use PR# to output the list to your printer.

Otherwise, write them down. Now, run ENTRY and create a song. Change all the parameters at the beginning of each part so they match the final parameters printed by the STATUS command. Now you can enter the second half of the song, and the various parameters (such as the envelope parameters and the transpose settings) will match whatever you left off with in the first half. When you've got the second half finished, delete any parameters at the beginning of each part that you haven't changed (in other words, that still match the final states from the STATUS command of the first half), save the song, and use PROCESS to append it to the first half.

Note that when the status command scans a part or subroutine, it does follow all subroutine calls to see what parameter changes might be present and to include the subroutine in the total time periods count. If subroutines are used to make the playback infinite for the part or subroutine being scanned, the final state list will not be printed, but the "subroutines used" list will be. (The total time periods will be listed as infinite.)

The **WIDTH** command is used to change the terminal width for a printer. You type WIDTH: and the width in printing columns. CAUTION: Apple's built-in ROM I/O routines function improperly with widths greater than 4Ø; memory may be erased if the width is set greater than 4Ø while the printer is PR#Ø. If the printer width is greater than 4Ø, always (1) set output to the printer using the PR# command, then (2) change the printing width with the WIDTH command. To go back to the Apple screen, always (3) change the printing width to 4Ø by typing WIDTH:4Ø and pressing return, then (4) set the printer to the screen by typing PR#Ø and pressing return.

PR# is not a PROCESS command, it is an Apple DOS command. The following DOS commands may be used from PROCESS: CATALOG, DELETE, FP, IN#, INT, LOCK, PR#, RENAME, and UNLOCK. Since Apple DOS commands use keyword separators at random (rather than consistently using a specific separator such as :), they cannot be abbreviated. PROCESS's use of : as a separator after all commands allows it to have a simple abbreviation routine. This routine lets you shorten your commands, if you wish, to the smallest number of letters needed to tell it apart from any other command. Or, you can add more letters if you like and abbreviate the word only slightly. Since PROCESS checks to see that all letters you do give are correct, there is no problem with a typing error like SATUS being taken as SAVE just because it starts with SA. If you type STAT you can be confident you're going to get "status". If you just type S, PROCESS will print AMBIGUOUS COMMAND so you'll know there are two or more commands which start with S, and you should add another letter or two. Some day, **all** quality software will incorporate similar human engineering. On the other hand, maybe all **quality** software already does.

The **AUXILIARY** command is used to load an auxiliary song from which parts and/or subroutines will be read to append to the main song. You type AUXILIARY: and the song name. You can load the same song you loaded as the main song if you wish. When the auxiliary song is loaded you can select various options. Typing 1 and pressing return gives you option 1 which just loads the auxiliary song.

Typing 2 and pressing return gives you option 2 which moves any "extra" subroutines from the auxiliary song to the main song. For example, if the main song has 3 subroutines (Ø-2) and the auxiliary song has 7 subroutines (Ø-6), subroutines 3 through 6 from the auxiliary song will be moved to the main song. Subroutines Ø though 2 in the auxiliary song remain in the auxiliary song. This is useful if you are adding the second half of a song (the auxiliary song) to the first half of a song (the main song) and you needed subroutines from the first half when entering the second half. (Perhaps the second half plays melodies which are already in subroutines in the first half.) Since you may have deleted all the notes in some unused subroutine to gain space for the second half, subroutines Ø-2 in the main song must stay as is. The new subroutines you've added for the second half (3-6 in this case) must be moved to the main song since they'll be needed. To finish the transfer, you'll need to append each part in the auxiliary song to the matching part in the main song.

If there are no subroutines in the main song, only options 1 and 2 will be available. If there are subroutines, you can also type 3 and press return for option 3. Option 3 will renumber all the subroutines in the auxiliary song and move them to the main song. For example, if the main song has 3 subroutines (Ø-2) and the auxiliary song has 7 (Ø-6), subroutines Ø-6 in the auxiliary song will be renumbered as 3-9 and moved to the main song. All calls in the auxiliary song will be renumbered too so they will still reference the proper subroutine. No subroutines will remain in the auxiliary song. This option is useful if you're adding the second half of a song (the auxiliary song) to the first half (the main song) and you didn't need any of the subroutines from the first half when entering the second half. Therefore, you'll want all your new subroutines shifted so they don't conflict with the first half's subroutines, and moved to the main song. Then you just append each part from the auxiliary song to the same part in the main song, and you've added the two halves together.

The **APPEND** command is used to move a part or subroutine from the auxiliary song (you must have already loaded one with the AUXILIARY command) to the end of a part or subroutine in the main song. You type APPEND: then a reference, then TO and then another reference. A "reference" is either PART and a part number, or SUB and a subroutine number. For example, to append part 3 from the auxiliary song to the end of part 2 in the main song, type APPEND:PART 3 TO PART 2 and press return. Since the APPEND command moves the specified part or

subroutine, once you move a given part or subroutine from the auxiliary song, it's no longer available to move again.

If the part or subroutine you try to append onto doesn't exist (in the main song), it will be created. Note that when a part is created, the usual parameters (KEY:C, TIME:4/4, and the various * items) that ENTRY would put in a new part are not created. This avoids the duplication that would occur when a part is moved into a new part. You must remember, however, to add all necessary parameters if you move a subroutine (which might not have them) to a new part.

CAUTION: if APPEND creates a new part, the stereo settings will be reset to standard settings.

One common use of APPEND is to change a part into a subroutine. For example, if you wish to change part 5 into a subroutine and your song already has 3 subroutines (Ø-2), you'll want to change part 5 into subroutine 3. To do this, you save your song (let's say you type SAVE:GALACTIC TRIUMPH while running ENTRY, then use FP to exit ENTRY). Type RUN PROCESS, and LOAD:GALACTIC TRIUMPH. Type AUXILIARY:GALACTIC TRIUMPH. Since your 3 subroutines are already present in the main song, you won't want to duplicate them by asking for option 3; so use option 1 to not move any subroutines (or, option 2 would do the same thing). Now type APPEND:PART 5 TO SUB 3 to change part 5 into subroutine 3. A subroutine 3 will be created in the main song (since there wasn't one before), and part 5 from the auxiliary song (which is also your GALACTIC TRIUMPH song) will be moved into the new subroutine. If you want to get rid of part 5 (since it's now also in subroutine 3), type DELETE:PART 5. If you want to keep both part 5 and subroutine 3, don't delete part 5. Now, type SAVE:NEW GALACTIC TRIUMPH, then type FP to exit PROCESS. Presto! Part 5 is now subroutine 3. If you discover you really meant to change part 4 into subroutine 3, then go back to PROCESS and start with LOAD:GALACTIC TRIUMPH again. Or, you could take NEW GALACTIC TRIUMPH and turn subroutine 3 into part 5, delete subroutine 3, append part 4 into subroutine 3, and delete part 4.

As just described, the **SAVE** command is used to save the main song after you've made the changes you want. Note that when you save a song, empty subroutines must be created for any subroutines that have been deleted but are still called. You will be given a chance to continue processing rather than save if this is the case. When you save a song, the main and auxiliary songs are cleared, and if you wish to continue processing the song you must load it again. Note that a song with zero parts cannot be saved.

The **FP** command is used to exit PROCESS.

Two additional commands are available for use by skilled programmers. (If you're a mere mortal, you won't need these commands.) They are **BLOAD** and **BSAVE.** BSAVE is the same as SAVE except it causes the main song to be saved as an Apple DOS "B" file in a special format. There are three formats available for songs. First, there's the "M:" format used by ENTRY and PLAY, where the song appears in the catalog as if it were an Integer BASIC program. This format allows the song to be readily copied from one disk to another, using Integer BASIC's LOAD and SAVE commands. The second format is the B format shown in the PERFORM section of this manual. It is the same as the "M:" format except the song appears in the catalog as a Binary file. The third format is the "B:" format available though the BSAVE command in PROCESS. It also appears in the catalog as a Binary file, but you can tell the three formats apart by their names. For example, GALACTIC TRIUMPH appears as M:GALACTIC TRIUMPH in the M: format, as GALACTIC TRIUMPH in the B format, and as B:GALACTIC TRIUMPH in the B: format.

The B: format is very similar to the B format. The only differences are: (1) there are always 18 bytes (9 pointers) after the part count byte (the first byte of the song data) even though the last pointers may not be used, (2) there are always 3 FE bytes before a subroutine, and (3) the relative addresses in CALL's are replaced by the subroutine number (Ø-99) in the third byte of the command. The B: format allows you to write your own programs which load the song and make various changes, then save the song again. The B: format song can then be loaded using the BLOAD command in PROCESS, and saved in the M: format using the SAVE command. The BLOAD command ignores the 18 bytes after the part count byte, and computes them by looking for the CHANNEL commands at the start of each part. Thus, you don't have to update the part pointers if you don't wish to. One thing you might forget is that the initial speed byte must be located by looking 16Ø bytes down from the end of the data, and not by just assuming the initial speed byte will be the first byte after the END command. (In other words, we might stick something between the END command and the initial speed byte while you're not looking.)

RATE

PROCESS was written before the RATE command was added to ENTRY. Thus, the RATE command will show as POKE 2Ø5,1,n (in the STATUS command, for example). Also, the "ALTERNATE SPEED" parameter is not used with this new version of ENTRY.

COMMANDS

APPEND:<part or sub> **TO** <part or sub>
AUXILIARY:<song>
BLOAD:<song>
BSAVE:<song>
CATALOG
CHANGE:<Ø-99> **TO** <Ø-99> [**IN** <subset>]
DELETE
DELETE:<part or sub>
FP
IN#
INT
LOAD:<song>
LOCK
PR#
RENAME
SAVE:<song>
STATUS
STATUS:<subset>
UNLOCK
WIDTH:<1-255>

<song> is a song name optionally followed by S<slot> and D<drive number> specifications (see your Apple DOS manual for S#D# details). <part or sub> is **PART**<Ø-8> or **SUB**<Ø-99>. <subset> is **ALL PARTS** or **PART**<Ø-8> or **SUB**<Ø-99>.

LINE 10

If you wish to change line 1Ø, LOAD PROCESS, list line 1Ø, change only the appropriate numbers, then SAVE PROCESS. Note that CARD=1 is for the MC1, and CARD=16 is for the MC16. Normally line 1Ø is set correctly by the "reconfigure programs" option in the HELLO program menu.

MLIST

The MLIST program is used to list a song in alphanumeric form on the screen, on a printer, or to a text file. To use the program, begin with RUN MLIST.

The **LOAD** command is used to select a song for listing and read it from the disk. You type LOAD: then the song name, and press return. (Note: do not type the "M:" part of the name. To load a song which appears as M:SOLFEGGIO in the catalog, type LOAD:SOLFEGGIO.) Once loaded, the song can be listed as many times as you like. If you wish to begin listing a different song, another LOAD command can be used to load in the next song.

The **LIST** command is used to list a song. To list all parts and all subroutines, just type LIST and press return. To list only the parts, type LIST:ALL PARTS (note: this can be abbreviate to LI:A if you prefer). To list a particular part, type LIST:PART and the part number. For example, LIST:PART 3 will list only part 3. Similarly, to list a particular subroutine, type LIST:SUB and the subroutine number, for example LIST:SUB 3. (These could be abbreviated to LI:P3 or LI:S3.)

You can also restrict the listing to a particular range of "command numbers". (Command numbers are explained in the PERFORM section, and in the SONG DATA FORMAT part of the ENTRY section.) All notes have command numbers from Ø to 191, and all rests have a command number of 192. All other commands have numbers from 193 to 255. Thus, if you restrict the listing to values from Ø to 192, you'll get only the notes and rests. If you restrict it to values from 193 to 255, you'll get everything else. The CALL command has a command number of 2Ø1, so if you wish to list only the subroutine calls you would restrict the listing to command number 2Ø1. This would be done by typing LIST:2Ø1. You could type LIST:2Ø1,192 to list both CALL commands and rests, for some unfathomable purpose. To list commands within a range of numbers, you use a dash (-). For example, LIST:Ø-191 would list all notes. If the starting number is to be Ø, or if the ending number is to be 255, the number can be omitted. LIST:Ø-191 is the same as LIST:-191. To list everything except notes and rests, type either LIST:193- or LIST:193-255. You can also specify several ranges. LIST:-191,193- would list everything except rests. Or, LIST:-191,193-2ØØ,2Ø2- would list everything except rests and subroutine calls.

When using a restricted range, you can also add the word IN and use the ALL PARTS, PART <Ø-8>, or SUB <Ø-99> options described above. For example, LIST:2Ø1 IN PART 2 will list only subroutine calls in part 2. (This could also be abbreviated LI:2Ø1IP2.)

Any list command can be followed by a star (*) if you wish to have an abbreviated listing. ATTACK:8192 prints as AT:8192 in an abbreviated listing. A sample command would be LIST:2Ø1 IN PART 2*. Note that when using just LIST, you must type LIST:* (not LIST*).

The **PR#** command is used to print the listing on an Apple "PR#"-compatible printer. For example, PR#1 will cause the output of all list commands to be sent to the printer plugged into slot 1. (All other output will continue to go to your present console.) If you wish to resume listing to the Apple screen, type PR#Ø. Since the PR# command is an Apple DOS command, it cannot be abbreviated.

To print the listing to an Apple text file, type PR# and the name of the file. For example, PR#SAMPLE will cause the output of all list commands to be written into a text file named SAMPLE. (Disk specifications such as PR#SAMPLE,D2 can be added if desired.) The output of as many list commands as you like can be written into the file. When you no longer wish to output lists to file, type PR#Ø to go back to the Apple screen, or type FP to exit MLIST.

The **WIDTH** command is used to adjust the width of your listing. The normal width is 4Ø characters, to match the Apple screen. CAUTION: Apple's built-in ROM I/O routines function improperly with widths greater than 4Ø, memory may be erased if a LIST command is used while PR# is Ø and the width is greater than 4Ø. Therefore, when selecting a larger width for a printer, always (1) set the printer slot using PR#, then (2) set the larger width using WIDTH:, then (3) make your listings. If you wish to list to the Apple screen now, first (4) set the width back to 4Ø, then (5) select the screen with PR#Ø.

The **FWIDTH** command is used for the same function as the WIDTH command, except it applies only to the output of LIST commands going to a text file. FWIDTH is normally 4Ø, but you may wish to make it larger to avoid breaking a long line into several strings.

The **FP** command is used to exit MLIST. It cannot be abbreviated.

Also, certain Apple DOS commands can be used from MLIST. They are: CATALOG, DELETE, FP, IN#, INT, LOCK, PR#, RENAME, and UNLOCK. Since Apple DOS commands use keyword separators at random (rather than consistantly using a specific separator such as :), they cannot be abbreviated.

Control-S can be used during a listing to temporarily suspend output. Pressing almost any key (except control-C or RESET) will resume the output. Control-C can be used to quit listing.

Notes and rests in the listing are preceded by a colon (:). The format of a note specification is described in the ENVELOPE section. The only difference is that a tied note or rest is printed in the same format as a rest, but with a T (for tied) instead of an R (for rest). Thus a note which displays in ENTRY as a D above middle C quarter note tied to a sixteenth note will be printed as :D3Q; :TS. Similarly, a quarter rest tied to a sixteenth rest prints as :RQ; TS. Non-standard durations are printed numerically.

Note that the S, F, and N specifications will match the graphics display of ENTRY rather than ENTRY's text display. That is, a note which is sharp due to a sharp in the key signature or a sharp note earlier in the measure will not be marked with an S. This also applies to flat notes and F.

Measure bars are not printed. Each measure begins a new line which starts with the measure number. When doing a restricted listing, measures with nothing to print are not printed.

SAMPLE LISTING

A sample listing of the beginning of "America" is shown below.

```
]RUN MLIST
-LOAD:AMERICA
-LIST
PART:Ø

1 KEY:1S; TIME:3/4; QUARTER:24Ø
  GAP:2Ø; TRANSPOSE:Ø; ATTACK:8192
  DECAY:25; VOLUME:55ØØØ; SUSTAIN:Ø
  RELEASE:15ØØ; :G3Q; :G3Q; :A4Q
2 :F3Q.; :G3E; :A4Q
3 :B4Q; :B4Q; :C4Q
4 :B4Q.; :A4E; :G3Q
5 :A4Q; :G3Q; :F3Q
6 :G3H.

PART:1

1 KEY:1S; TIME:3/4; QUARTER:24Ø
  GAP:2Ø; TRANSPOSE:Ø; ATTACK:8192
  DECAY:25; VOLUME:55ØØØ; SUSTAIN:Ø
  RELEASE:15ØØ; :D3Q; :D3Q; :E3Q
2 :D3Q.; :E3E; :F3Q
```

```
3 :G3Q; :G3Q; :G3Q
4 :G3Q.; :F3E; :E3Q
5 :E3Q; :D3Q; :D3Q
6 :D3H.

PART:2

1 KEY:1S; TIME:3/4; QUARTER:24Ø
  GAP:2Ø; TRANSPOSE:Ø; ATTACK:8192
  DECAY:25; VOLUME:55ØØØ; SUSTAIN:Ø
  RELEASE:15ØØ; :G2Q; :E2Q; :C2Q
2 :D2Q; :C2Q; :D2Q
3 :G2Q; :E2Q; :C2Q
4 :D2Q; :DS2Q; :E2Q
5 :C2Q; :D2Q; :D2Q
6 :G2H.

-LIST:193-*
PART:Ø

1 K:1S; TI:3/4; Q:24Ø; GA:2Ø; TR:Ø
  AT:8192; DEC:25; V:55ØØØ; SUS:Ø
  R:15ØØ

PART:1

1 K:1S; TI:3/4; Q:24Ø; GA:2Ø; TR:Ø
  AT:8192; DEC:25; V:55ØØØ; SUS:Ø
  R:15ØØ

PART:2

1 K:1S; TI:3/4; Q:24Ø; GA:2Ø; TR:Ø
  AT:8192; DEC:25; V:55ØØØ; SUS:Ø
  R:15ØØ
```

The second listing shows an abbreviated (*) listing of everything except notes and rests.

COMMANDS

CATALOG
DELETE
FP
FWIDTH:<1-255>
IN#
INT
LIST[:*]
LIST:[<numbers> **IN**]<subset>[*]
LIST:<numbers>[*]
LOAD:<song>
LOCK
PR#
PR#<file name> (cannot be abbreviated)
RENAME
UNLOCK
WIDTH:<1-255>

<numbers> is a number range, or several ranges separated by commas. A number range is a number, or a dash and a number, or a number and a dash, or a number and a dash and another number. A number is an integer from Ø to 255. <subset> is **ALL PARTS** or **PART**<Ø-8> or **SUB**<Ø-99>. <song> is a song name optionally followed by S<slot> and D<drive number> specifications (see your Apple DOS manual for S#D# details).

RATE

MLIST was written before the RATE command was added to ENTRY. Thus, the RATE command will list as POKE 2Ø5,1,n (where "n" is the selected rate).

5
PLAY/DISCO/HUSTLE

PLAY

The PLAY program is used to play songs entered with ENTRY. 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 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 or HUSTLE programs or custom files.

Normally, PLAY will be set for the correct slot by the boot-up configuration program. However, if you wish to change line 1Ø yourself, LOAD PLAY, LIST 1Ø, and carefully retype the line changing only the slot/units digit(s). Now type SAVE PLAY.

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 FP should always be used after a STOP command.

FP
FP is used to stop using PLAY. The PLAY program is erased and must be reloaded if you desire to run it again.

RESET may be pressed to stop song playback. When the] prompt appears, type RUN. Do not use RESET while loading a song, and never use control-C.

ALBUM FILES

The DISCO and HUSTLE programs are used to create "album files". An album file is a text file containing the names of songs to be played. Album files can be used in two ways: to play several songs in a predetermined order, or to play several songs in a random order. When using a random playback order, a particular song can be designated as the song to always play first, and likewise another song can always be played last.

Album files can be created with either the DISCO program or the HUSTLE program. Once created, all songs can be played in the predetermined order by typing EXEC ALBUM. The songs can be played in random order by typing RUN DISCO. Note that the album files contain only the names of the songs to be played (in a special format as required by the PLAY program) and not the actual songs themselves. The songs, and the PLAY program, must be stored on the disk the album file is stored on.

DISCO

The DISCO program is used mainly to change the order of the names in the album file so playback order will be random. The disk containing the album file (and the DISCO and PLAY programs, and songs) must not be write-protected. (When using EXEC ALBUM for non-random playback, the disk can be write-protected.) The random order playback function is requested simply by typing RUN DISCO (an album file must already exist).

DISCO can also be used to create an album file. If an album file already exits, begin by typing DELETE ALBUM. Then type LOAD DISCO and RUN 1ØØØ. DISCO will print a brief set of instructions. You simply type in the name of each song to be played. If you won't be using randomized playback order, type the names in the order you wish them to be played. You can type CATALOG if you wish to see the catalog. If you wish to have a song which will always be played first (when using randomized order), the first song name you type must be START. If you wish to have a song which will always be played last, it must be entered last and be named END. If you do not wish to have an "END" song, type STOP when you are finished entering names.

If the songs to be played are on two disks, all song names must be followed with ,D1 or ,D2 (whichever is appropriate). If your system has more than two drives, you can create album files which use songs on several drives. In this case, all song names must be followed with the slot and drive numbers (e.g. ,S6,D1).

To add song names to an album file, type LOAD DISCO and RUN 2ØØØ. Songs are

added in the same fashion as just described. If an END song was previously entered, it will remain at the end of the album file. If a START song is added, it may not be played first since it may not be the first song listed.

HUSTLE

The HUSTLE program is used to create an album file containing the names of all songs on the disk, or to delete particular song names from an existing album file.

Note that although the HUSTLE program has a line 1Ø which reads 1Ø CARD=1 (which indicates the program is configured for an MC1), this line never needs to be changed. Even when using an MC16 (which would imply a CARD=16 setting), this line should be set with CARD=1 because the older MC16-format album file is no longer used.

If you wish to create a new album file and an old album file already exists on the disk, begin by typing DELETE ALBUM. Then type RUN HUSTLE. If you wish to add or delete songs from an existing album file, just type RUN HUSTLE.

Instructions for using HUSTLE are presented as the HUSTLE program runs. Note that you should stop the HUSTLE program only by using the exit option. Otherwise a temporary file, named ALBUMX, may be left on your disk.

When creating an album file using two or more disks of songs, drive specifications will automatically be included in the album file (e.g. ,D1 and ,D2). Therefore, the disks must be placed in the same drives during playback as they were in during album file creation.

PLAYING THE ALBUM

To play the whole sequence of songs after creating your album file (or with an album file supplied on your MC1/MC16 software disk or song disk), just type EXEC ALBUM. Or, for randomized playback order, type RUN DISCO (your disk must not be write-protected to use DISCO). A properly configured PLAY program must be on the disk the album file is on. When album playback is complete, you can type RUN to continue using the PLAY program (do <u>not</u> type RUN PLAY), otherwise type FP. If you wish to hear the songs again, type FP and then EXEC ALBUM (or RUN DISCO).

6
MC16 CHROMA

The CHROMA subroutine is used to write your own programs which use the MC16 to produce chromatic (equal tempered) pitches. (CHROMA is not supplied with the MC1.) The various routines in CHROMA are:

1. **INITIALIZER.** Written in BASIC, this routine initializes the music card, the CHROMA routine, and the PULSE routine.
2. **PARTIAL INITIALIZER.** Written in BASIC, this routine is used to initialize additional music cards.
3. **CHROMA.** Written in 65Ø2 assembly language, this routine is used to program "normal mode" (square wave) pitches.
4. **PULSE.** Written in 65Ø2 assembly language, this routine is used to program "pulse mode" (pulse wave) pitches.

The parameters required by these routines, their calling procedures, functions, and results are described below.

INITIALIZER

The INITIALIZER uses the value of the variable SLOT. Prior to calling the INITIALIZER, this variable should be set to the expansion slot number one of your music cards is plugged into. The INITIALIZER is called using GOSUB 32767. It will initialize the music card, correct memory addresses in the CHROMA and PULSE routines, assign values to the variables PITCH and VOLØ, and poke SLOT*16 at PITCH+2 and Ø at PITCH+3 (see table below). "Initialize the music card" means to set all three channels to zero volume and "normal mode".

POKE ADDRESS	NAME	DESCRIPTION
PITCH	PITCH	Pitch number
PITCH+1	PART	Channel (part) number
PITCH+2		Slot number times 16
PITCH+3	OFFSET	Pitch offset
PITCH+4	WIDTH	Pulse width
PITCH+5		Divisor low
PITCH+6		Divisor high
PITCH+7	CHROMA	CHROMA entry point
(PITCH+8 and PITCH+9 are reserved.)		
PITCH+1Ø	PULSE	PULSE entry point
(PITCH+11 and PITCH+12 are reserved.)		
PITCH+13		(start of divisor table)

The table above shows the memory locations used for parameter storage by the CHROMA and PULSE routines. The address of this table is indicated by the value assigned to PITCH, which is based on the value of HIMEM (or the length of your

program when using Applesoft). Note that when using Integer BASIC, HIMEM must not be -32498, -32433, or any value in between.

The variable VOLØ is used to set volume levels and change modes.

POKE ADDRESS	NAME	DESCRIPTION
VOLØ	VOLØ	Volume for channel Ø
VOLØ+1	VOL1	Volume for channel 1
VOLØ+2	VOL2	Volume for channel 2
VOLØ+3		Mode control A
VOLØ+7		Mode control B

Values poked at the above addresses go directly to the music card and cause the volume or mode to change immediately. Values from Ø to 255 can be poked for volume (Ø=off or 1=soft to 255=loud). The following values can be poked for mode control (other values should not be used).

POKE ADDRESS	VALUE	FUNCTION
VOLØ+3	Ø	Both channels Ø and 1 to pulse mode
VOLØ+3	1	Channel Ø to normal mode, channel 1 to pulse mode
VOLØ+3	2	Channel Ø to pulse mode, channel 1 to normal mode
VOLØ+3	3	Both channels Ø and 1 to normal mode
VOLØ+7	5Ø	Channel Ø to pulse mode
VOLØ+7	54	Channel Ø to normal mode
VOLØ+7	114	Channel 1 to pulse mode
VOLØ+7	118	Channel 1 to normal mode
VOLØ+7	182	Channel 2 to normal mode (used by the INITIALIZER)

The INITIALIZER and PARTIAL INITIALIZER set all three channels to normal mode. To change modes, set the mode by poking the value shown above to VOLØ+7, then the appropriate value (above) to VOLØ+3.

The value assigned to VOLØ by the INITIALIZER or PARTIAL INITIALIZER is different for each expansion slot and is calculated by the formula VOLØ=SLOT*16-16256.

The mnemonic variable names shown in the first table can be set using the following statements. (Note: the variable name PART was given as CHANNEL, which is more appropriate, in previous manuals. However, Applesoft does not allow two variables to be named CHANNEL and CHROMA.) The setup and calling of the INITIALIZER is included:

```
1Ø SLOT=4      (replace 4 with the proper slot number)
2Ø GOSUB 32767 : PART=PITCH+1 : OFFSET=PITCH+3 : WIDTH=PITCH+4 :
   CHROMA=PITCH+7 : PULSE=PITCH+1Ø : VOL1=VOLØ+1 : VOL2=VOLØ+2
```

NOTE: Applesoft does not allow three variables to be named VOLØ, VOL1, and VOL2. Applesoft users should pick names for VOL1 and VOL2 (if they need these variables) which do not begin with the same 2 letters as any other variable, and

complain to Microsoft.

PARTIAL INITIALIZER

When more than one music card is used, the units not initialized with the INITIALIZER (GOSUB 32767) must be initialized with the PARTIAL INITIALIZER. For each additional card, set SLOT to the proper expansion slot number, and call the PARTIAL INITIALIZER using GOSUB -2. It will initialize the music card and set VOLØ to the volume control address for that slot number. Previous values of VOLØ set by the INITIALIZER or PARTIAL INITIALIZER should be assigned to other variables if they must be retained. (The value of VOLØ for any slot is computed by the formula VOLØ=SLOT*16-16256.) Note that GOSUB -2 does not cause the slot number times 16 to be written at PITCH+2 or a zero to be written at PITCH+3. GOSUB -3 can be used instead if you wish to have these values poked. (On systems where Applesoft doesn't allow GOSUB with negative numbers, use 63998 instead of -2 and 63997 instead of -3.)

CHROMA

CHROMA uses the parameters poked at PITCH, PART, PITCH+2, and OFFSET. It changes the contents of PITCH+5 and PITCH+6. When called using CALL CHROMA (or CALL PITCH+7), CHROMA programs the desired channel (indicated by PART) on the desired music card (indicated by the slot number times 16 at PITCH+2) with the desired pitch (indicated by PITCH and OFFSET). To do this, CHROMA will calculate a two-byte divisor which it stores at PITCH+5 and PITCH+6 in case it is needed for PULSE (see the PULSE routine in this section). The precise function of these poked parameters is as follows:

PART (PITCH+1)

This indicates which of the three channels is to be programmed. It must be an integer from Ø to 2. Adding 128 will inhibit programming of the music card but the divisor will still be computed and stored.

PITCH+2

This indicates the slot number of the music card to be programmed. The value poked must be the slot number (Ø to 7) times 16. If only one music card is used, this parameter does not need to be poked since it is initialized to SLOT*16 by the INITIALIZER.

PITCH

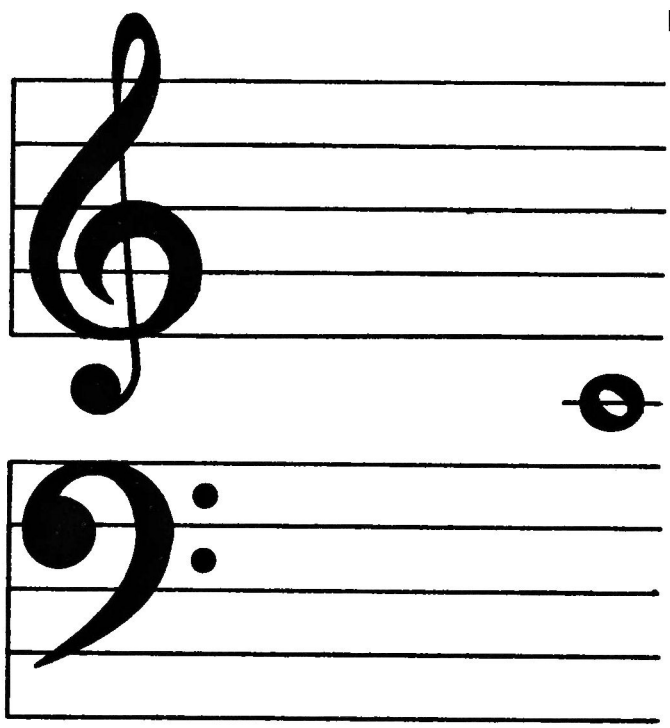
This indicates the quarter-tone pitch to be programmed. The values for half-tones in the lowest octave are:

Ø	A	8	C sharp	16	F
2	A sharp	1Ø	D	18	F sharp
4	B	12	D sharp	2Ø	G
6	C	14	E	22	G sharp

For quarter-tones, add 1. For higher octaves, add the numbers shown below to the numbers shown above. The frequency of the A in that octave is also shown below. (Note: "octaves" here start at A.)

A (Hz)	Add	A (Hz)	Add	A (Hz)	Add	A (Hz)	Add
27.5	Ø	11Ø	48	44Ø	96	176Ø	144
55	24	22Ø	72	88Ø	12Ø	352Ø	168

The highest pitch (G sharp plus a quarter-step) in the highest octave is 22+1+168 (or 191), so pitch values should be from Ø to 191. Some common notes and their values are (for sharp, add 2; for flat, subtract 2):



Hex	Decimal	Note
7Ø	112	F
6E	11Ø	E
6A	1Ø6	D
66	1Ø2	C
64	1ØØ	B
6Ø	96	A 44Ø
5C	92	G
58	88	F
56	86	E
52	82	D
4E	78	Middle C
4C	76	B
48	72	A
44	68	G
4Ø	64	F
3E	62	E
3A	58	D
36	54	C
34	52	B
3Ø	48	A
2C	44	G

OFFSET (PITCH+3)

This indicates how sharp the pitch should be from standard tuning. Ø is used for standard A=44Ø Hz tuning (as initialized by GOSUB 32767 or GOSUB -3), and numbers from 1 to 255 are used to raise the pitch slightly. All pitches selected using OFFSET are less than or equal to the pitch selected by a PITCH setting one higher. Note that the pitches selected by various values of PITCH increase exponentially, whereas the pitches selected by various values of OFFSET (with a constant PITCH setting) increase linearly.

PULSE

The PULSE routine is used to create pulse waves using channel Ø and/or channel 1. The frequency (pitch) of the pulse wave will be the same as the frequency of channel 2. The INITIALIZER sets all channels to normal mode, so channels to be used with PULSE must be changed to "pulse mode" as previously described. The parameters poked at PART, PITCH+2, WIDTH, PITCH+5, and at PITCH+6 are used. PULSE is called using CALL PULSE (or CALL PITCH+1Ø). The precise function of each parameter is as follows:

PART (PITCH+1)
This indicates which of the two channels is to be programmed. It must be either Ø or 1. Adding 128 will inhibit programming of the music card but the divisor will still be calculated and stored (see divisor storage locations below).

PITCH+2
This indicates the slot number of the music card to be programmed. The value must be the slot number (Ø to 7) times 16.

WIDTH (PITCH+4)
This indicates the width of the low part of each cycle. Numbers from Ø to 126 indicate a short low portion, and numbers from 128 to 255 indicate a long low portion. 127 is used to program a square waveform.

PITCH+5 and PITCH+6
These must contain the divisor currently programmed for channel 2. If CHROMA was called most recently for channel 2, these locations will already be set to the divisor (by CHROMA).

The divisor calculated by PULSE is stored at locations 81 and 82 decimal (61 and 62 in Applesoft). It may be read using peek immediately after calling PULSE.

CHROMA EXAMPLE

To program a three note chord of Middle C, E, G at maximum volume, begin by loading CHROMA. Now type in the following program, remembering to change the 4 to the correct expansion slot number.

```
1Ø SLOT=4
2Ø GOSUB 32767 : PART=PITCH+1 : OFFSET=PITCH+3 : CHROMA=PITCH+7
3Ø POKE PART,Ø : POKE PITCH,78 : CALL CHROMA : POKE VOLØ,255
4Ø POKE PART,1 : POKE PITCH,86 : CALL CHROMA : POKE VOLØ+1,255
5Ø POKE PART,2 : POKE PITCH,92 : CALL CHROMA : POKE VOLØ+2,255
6Ø END
```

Now run the program. The music card will be programmed for the C E G chord, and it will continue to produce the chord until programmed to do something else. The chord can be cleared by typing GOTO -2.

PULSE EXAMPLE

The following program produces one tone with the pitch controlled by Paddle Ø and the pulse width controlled by Paddle 1. As in the above example, begin by loading CHROMA. Then add the program below, remembering to correct the slot number.

```
1Ø SLOT=4
2Ø GOSUB 32767 : PART=PITCH+2 : WIDTH=PITCH+4
3Ø CHROMA=PITCH+7 : PULSE=PITCH+1Ø : POKE VOLØ+7,5Ø : POKE VOLØ+3,2
4Ø POKE PART,2 : POKE PITCH,PDL(Ø)/2 : CALL CHROMA
5Ø POKE PART,Ø : POKE WIDTH,PDL(1) : CALL PULSE
6Ø POKE VOLØ,255 : GOTO 4Ø
```

Now run the program, and twist the paddle knobs like mad. Stop the program, and type POKE VOLØ,Ø to stop the noise.

7
MC1 BARE HANDED

The MC1 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 music card. 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 music card and P is the stereo position number (Ø=left, 1=right, and 2=middle).

INITIALIZATION

Before use, the music card should be initialized. Upon power on, the music card 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.

6502 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).

8
MC16 BARE HANDED

The MC16 is programmed by means of 8 "ports". Each port is assigned a particular memory address, and information can be sent to a port by writing a byte (an integer from Ø to 255) to that memory address (using 65Ø2 Assembly Language or BASIC's POKE). Reading from these memory addresses does not affect the music card. The ports are numbered from Ø to 7. 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 music card and P is the desired port number (both should be Ø to 7).

The function of each port is as follows:

PORT	FUNCTION
Ø	Volume control for channel Ø
1	Volume control for channel 1
2	Volume control for channel 2
3	Mode control A
4	Divisor for channel Ø
5	Divisor for channel 1
6	Divisor for channel 2
7	Mode control B

Ports Ø-2 are used to control the volume. A byte written to one of these ports will cause the volume of the appropriate channel to change immediately to the new value (Ø=off or 1=soft to 255=loud). The relative output voltage for any volume setting (VOL) is computed by 2 (VOL/32)*(VOL MOD 32 + 33)-33 with Integer BASIC, or by 2 INT(VOL/32)*(VOL-INT(VOL/32)*32 + 33)-33 with Applesoft BASIC.

Ports 3 and 7 are used for mode control. Before use, all channels must be initialized to either normal mode or pulse mode to insure proper operation. Port 3 selects whether the pitch control will be provided by the Apple or by the output of Channel 2. Port 7 selects whether the divisor will control the pitch or the pulse width. Normally both ports 3 and 7 are set to indicate either normal mode or pulse mode. Port 7 should be programmed before port 3 for best results.

The value written to port 3 has the following effects:

VALUE	MEANING
Ø	Both channels Ø and 1 to pulse mode
1	Channel Ø to normal mode, channel 1 to pulse mode
2	Channel Ø to pulse mode, channel 1 to normal mode
3	Both channels Ø and 1 to normal mode

Other values should not be used.

Values written to port 7 have the following effects:

VALUE	MEANING
5Ø	Set channel Ø to pulse mode, channels 1 and 2 not affected
54	Set channel Ø to normal mode, channels 1 and 2 not affected
114	Set channel 1 to pulse mode, channels Ø and 2 not affected
118	Set channel 1 to normal mode, channels Ø and 2 not affected
182	Set channel 2 to normal mode, channels Ø and 1 not affected

Other values should not be used except as noted in the TIMING MODE section.

When a channel is set to a mode using port 7, the output of its pitch generator will go high and stay high until both bytes of a divisor are written. The high part of the cycle will then begin. (Note: port 3 should be set after port 7 is set but before the first divisor is programmed.)

When a channel is set to pulse mode with port 7 but normal mode with port 3, the output of its pitch generator will stay high. When a channel is set to pulse mode with port 3 but normal mode with port 7, the output of its pitch generator will be high when the output of channel 2's pitch generator is low, and when the channel 2 output goes high the mixed-mode channel will begin normal square wave operation starting with the high part of the cycle. (Once the channel 2 output returns to low, the mixed-mode channel will go high and stay high until the channel 2 output goes high again.)

Any of the three channels can also be set to a special "timing mode" where the channel is used to simulate the Apple "paddle" timers, but with a programmable setting. See the TIMING MODE section for details.

Ports 4-6 are used to program the divisor. Once a channel has been initialized, it will be expecting the low byte of the divisor (D MOD 256). Once the low byte is written, it will be expecting the high byte of the divisor (D/256). Once the high byte is written, the new divisor will be used by the pitch generator; and the low byte of the next divisor will be expected.

When in normal mode, the divisor determines the frequency to be produced by the pitch generator. The duty cycle is always approximately 5Ø% and cannot be altered. The output frequency will be 1782ØØØ/D Hz (where D is the divisor programmed) plus or minus Ø.Ø15%. The value D must be an integer from 32 to 65536. (Note: 65536 must be programmed as Ø. Values less than 32 are possible but should not be used.) When a new divisor is programmed, it does not take effect until the associated pitch generator's output goes high after the high byte of the divisor was written.

When in pulse mode, the divisor determines the time duration of the low portion

of the pulse wave. The frequency is determined by the frequency output of channel 2's pitch generator. Just after the low to high change of channel 2's pitch generator output, the output of the pulse mode channel's pitch generator will go low. It will stay low for D/1782ØØØ seconds plus or minus Ø.Ø15%. If the channel 2 output has again gone high during this time, the pulse mode output will stay low. Otherwise, the pulse mode output will go high and stay high until the next time the channel 2 output goes high. The value D must be an integer from 1 to 65536. (Note: 65536 must be programmed as Ø.) When a new divisor is programmed, it does not take effect until the first low to high change in the output of channel 2's pitch generator after the high byte of the divisor was written.

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(1782ØØØ/(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.

TUNING

It may be useful to know that musicians use "cents" to express the amount of deviation from correct tuning for half tones. A note too high (sharp) by 1ØØ cents would be the right frequency for the next higher half step. The formula for cents is (12ØØ*LOG(F/X))/LOG(2) where X is the correct frequency in Hz and F is the actual frequency produced in Hz. (The LOG may be in any base, as long as it is always the same base.) Inaccurate tuning in the music card's pitches results mainly from the fact that only integral values can be used for the divisor (D). This creates pitches out of tune by amounts varying from Ø to Ø.Ø2Ø cents in the lowest 12 notes of the piano scale, which increase to Ø.Ø67 to 1.2Ø4 cents in the top 12 notes. (The Ø.Ø15% crystal accuracy adds a maximum of Ø.26Ø cents.) Tuning accuracy within 2 cents should be considered excellent and suitable for any purpose.

9
PERFORM (& FLASH)

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

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 (and UNITS setting, for the MC16) 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 music card.) The following Applesoft BASIC program converts songs into binary files. Type it in and save it.

```
10 POKE 76,PEEK(115) : POKE 77,PEEK(116) : POKE 217,0
20 HIMEM:3000 : INPUT "SONG NAME?";A$
30 POKE PEEK(54)+PEEK(55)*256+3065,0
40 PRINT CHR$(4)"LOADM:"A$ : A=PEEK(202)+PEEK(203)*256
50 L=PEEK(76)+PEEK(77)*256-A : PRINT CHR$(4)"BSAVE"A$",A"A",L"L
60 PRINT "LENGTH: "L : PRINT CHR$(4)"FP"
```

To use the program, begin by typing FP. 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 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 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, 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 FP to exit ENTRY. You are now ready to convert the song. Type FP and 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,A8192 and then POKE 36864,Ø (return) POKE 36865,32 (return) and POKE 6,1Ø5 (return). Type CALL 36866 to play the song. When playback is finished, you could play the song again just by typing CALL 36866.

What are the mystic pokes for? Locations 36864 and 36865 must be set to the starting memory address of the song data. We loaded the song at 8192. Note that 32*256+Ø (32 and Ø being the numbers we poked) is 8192, the starting address. At location 6, we poked the initial playback rate. The initial speed of DIXIE BOOGIE is 16Ø and it has 5 parts, so the initial rate is 16Ø-5*11, or 1Ø5.

With a few precautions, you could have a BASIC program do the BLOADs, POKEs, and CALL. 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.

PERFORM starts at location 36864 ($9ØØØ) and ends at about 38399 ($95FF). This area is easily protected by using HIMEM:36864 as the first line in your Applesoft program. However, you'll probably want to use a smaller HIMEM value so you can create a protected area for loading songs into. For example, HIMEM:8192 would allow you to load a 28K song. If you won't be loading songs this large, a larger HIMEM value should be used so your Applesoft program can be larger. Using a variable for the HIMEM setting makes it easier to adjust, for example:

```
1Ø ADRS=8192: HIMEM:ADRS
2Ø PRINT CHR$(4)"BLOAD PERFORM"
3Ø PRINT CHR$(4)"BLOAD DIXIE BOOGIE,A"ADRS
4Ø POKE 36864,ADRS-INT(ADRS/256)*256
5Ø POKE 36865,ADRS/256
6Ø POKE 6,PEEK(PEEK(43616)+PEEK(43617)*256+ADRS-161)-11*PEEK(ADRS)
7Ø CALL 36866
```

The only other detail is that in this example we used ENTRY to initialize the music card (when DIXIE BOOGIE was configured), and for general purpose BASIC programs you would probably want to have your program initialize the music card. 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.

MUSIC CARD INITIALIZATION

If you have a line which sets SLOT (and UNITS, for an MC16), like the one in

ENTRY or PLAY, you can use this variable in an initialization routine for the music card. Generally, any program which uses the music card should have this initialization routine near the beginning. For the MC1, use:

```
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
```

For the MC16, use:

```
FOR S=SLOT TO SLOT+UNITS-1
PN=16*S-16256 : POKE PN,Ø : POKE PN+1,Ø : POKE PN+2,Ø
POKE PN+3,3 : POKE PN+7,54 : POKE PN+7,118 : POKE PN+7,182
NEXT S
```

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 music card initialization routine (above). It also needs the variable ADRS set to the starting address of the song to be configured.

```
FOR B=1 TO PEEK(ADRS)
PNTR=PEEK(B+B+ADRS-1)+PEEK(B+B+ADRS)*256+ADRS
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
```

For the MC16, use:

```
FOR B=1 TO PEEK(ADRS)
PNTR=PEEK(B+B+ADRS-1)+PEEK(B+B+ADRS)*256+ADRS : CHAN=B-1
IF UNITS>1 THEN CHAN=PEEK(PNTR+2)/(1+15*(3-UNITS))
CHAN=CHAN-INT(CHAN/16)*16 : POKE PNTR+1,INT(CHAN/4)*12+CHAN+SLOT*16
NEXT B
```

You might wish to add:

```
POKE 36864,ADRS-INT(ADRS/256)*256 : POKE 36865,ADRS/256
POKE 6,PEEK(PEEK(43616)+PEEK(43617)*256+ADRS-161)-11*PEEK(ADRS)
CALL 36866 : RETURN
```

to the end of either version in order to create a subroutine which can be GOSUBed in order to configure and play the song at address ADRS.

READING THE "INITIAL SPEED"

Assuming the song was just loaded using PRINT CHR$(4)"BLOAD song name,A"ADRS the initial speed of an ENTRY-created song can be read into the variable S with the following statement: S=PEEK(PEEK(43616)+PEEK(43617)*256+ADRS-161). (43616 is the DOS 3.2/3.3 BLOAD program length address for 48K and larger systems.) This can be converted to initial rate by subtracting 11 times the number of parts. PEEK(ADRS) gives the number of parts.

SAMPLE PERFORM SESSION

In the following sample session, we write a small program which plays a song, much like the PLAY program does. The program is for the MC1, changes needed for the MC16 are shown in parentheses.

```
]FP
]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 CHR$(4)"LOADM:"A$ : A=PEEK(2Ø2)+PEEK(2Ø3)*256
]5Ø L=PEEK(76)+PEEK(77)*256-A : PRINT CHR$(4)"BSAVE"A$",A"A",L"L
]6Ø PRINT "LENGTH: "L : PRINT CHR$(4)"FP"
]SAVE CONVERT
]RUN
SONG NAME?DIXIE BOOGIE

LENGTH: 3298

]FP
]5 ADRS=8192 : HIMEM:ADRS
]1Ø SLOT=4    (]1Ø SLOT=4 : UNITS=1)
]2Ø PRINT CHR$(4)"BLOAD PERFORM"
]3Ø PN=SLOT*16-16256    (]3Ø FOR S=SLOT TO SLOT+UNITS-1)
]4Ø FOR P=PN TO PN+2
    (]4Ø PN=16*S-16256 : POKE PN,Ø : POKE PN+1,Ø : POKE PN+2,Ø)
]5Ø POKE P,159 : POKE P,191 : POKE P,223 : POKE P,255 : POKE P,231
    (]5Ø POKE PN+3,3 : POKE PN+7,54 : POKE PN+7,118 : POKE PN+7,182)
]6Ø NEXT P    (]6Ø NEXT S)
]7Ø INPUT "SONG NAME?";A$ : PRINT CHR$(4)"BLOAD"A$",A"ADRS
]8Ø FOR B=1 TO PEEK(ADRS)
]9Ø PNTR=PEEK(B+B+ADRS-1)+PEEK(B+B+ADRS)*256+ADRS
```

```
    (]9Ø PNTR=PEEK(B+B+ADRS-1)+PEEK(B+B+ADRS)*256+ADRS : CHAN=B-1)
]1ØØ CHAN=PEEK(PNTR+2)-INT(PEEK(PNTR+2)/4)*4
    (]1ØØ IF UNITS>1 THEN CHAN=PEEK(PNTR+2)/(1+15*(3-UNITS))   )
]11Ø NAHC=INT((PEEK(PNTR+2)-INT(PEEK(PNTR+2)/16)*16)/4)
    (line 11Ø not needed for MC16)
]12Ø IF NAHC THEN NAHC=3-NAHC
    (]12Ø CHAN=CHAN-INT(CHAN/16)*16 : POKE PNTR+1,INT(CHAN/4)*12+CHAN+SLOT*16)
]13Ø POKE PNTR+1,SLOT*16+CHAN*4+NAHC
    (line 13Ø not needed for MC16)
]14Ø NEXT B
]15Ø POKE 36864,ADRS-INT(ADRS/256)*256 : POKE 36865,ADRS/256
]16Ø POKE 6,PEEK(PEEK(43616)+PEEK(43617)*256+ADRS-161)-11*PEEK(ADRS)
]17Ø CALL 36866 : GOTO 7Ø
]SAVE YALP
]RUN
SONG NAME?DIXIE BOOGIE

(song plays)
```

FLASH

The FLASH program is the same as the PERFORM program except it generates the lo-res color graphics display (like ENTRY and PLAY). It is used in exactly the same fashion as PERFORM, except that the color display must be set up before FLASH is called. The following routine sets up the display:

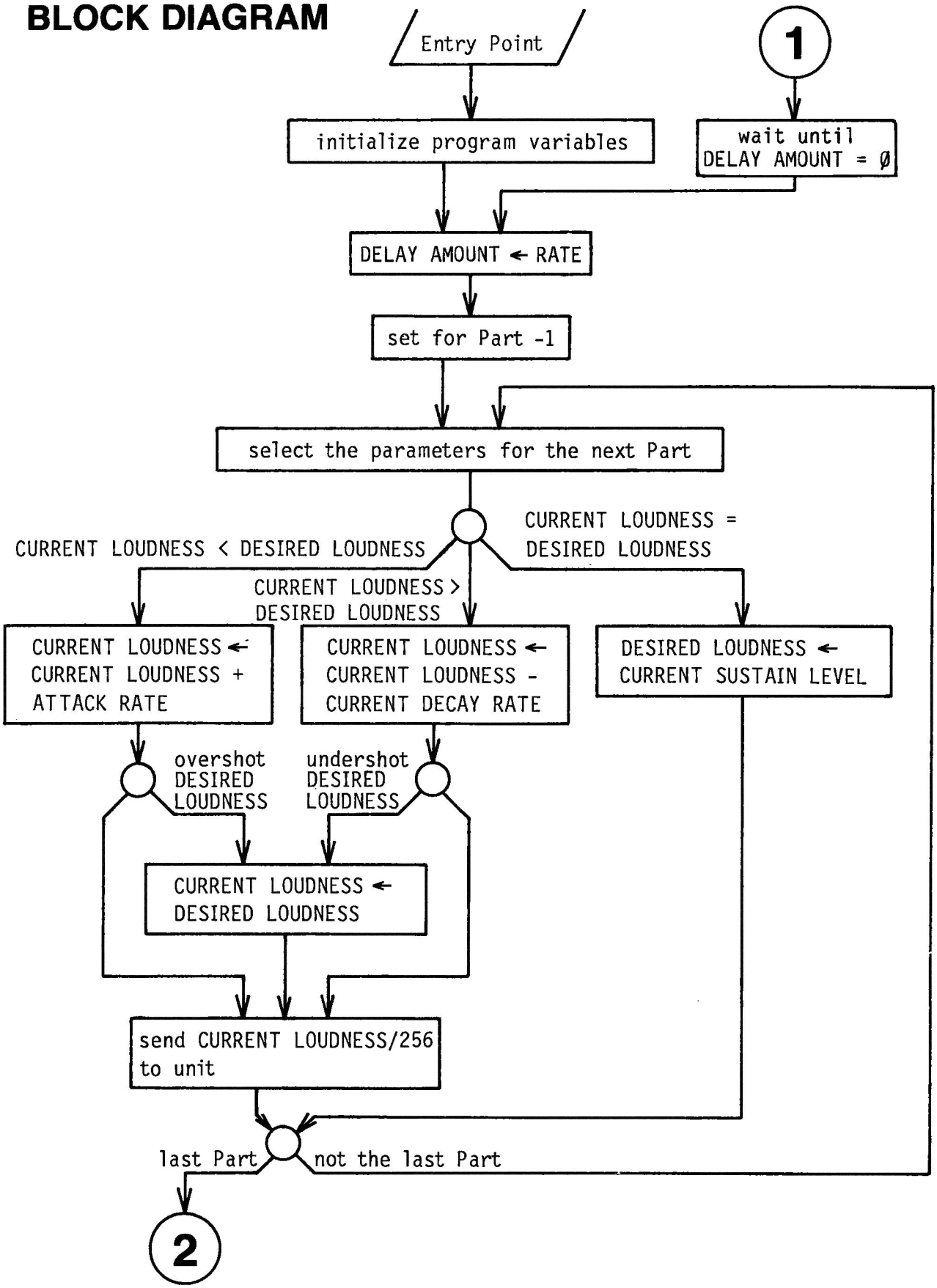
```
GR
FOR A=1 TO PEEK(ADRS)
COLOR=2 : HLIN Ø,39 AT A*4-2
COLOR=13 : PLOT 15,A*4-2
NEXT
```

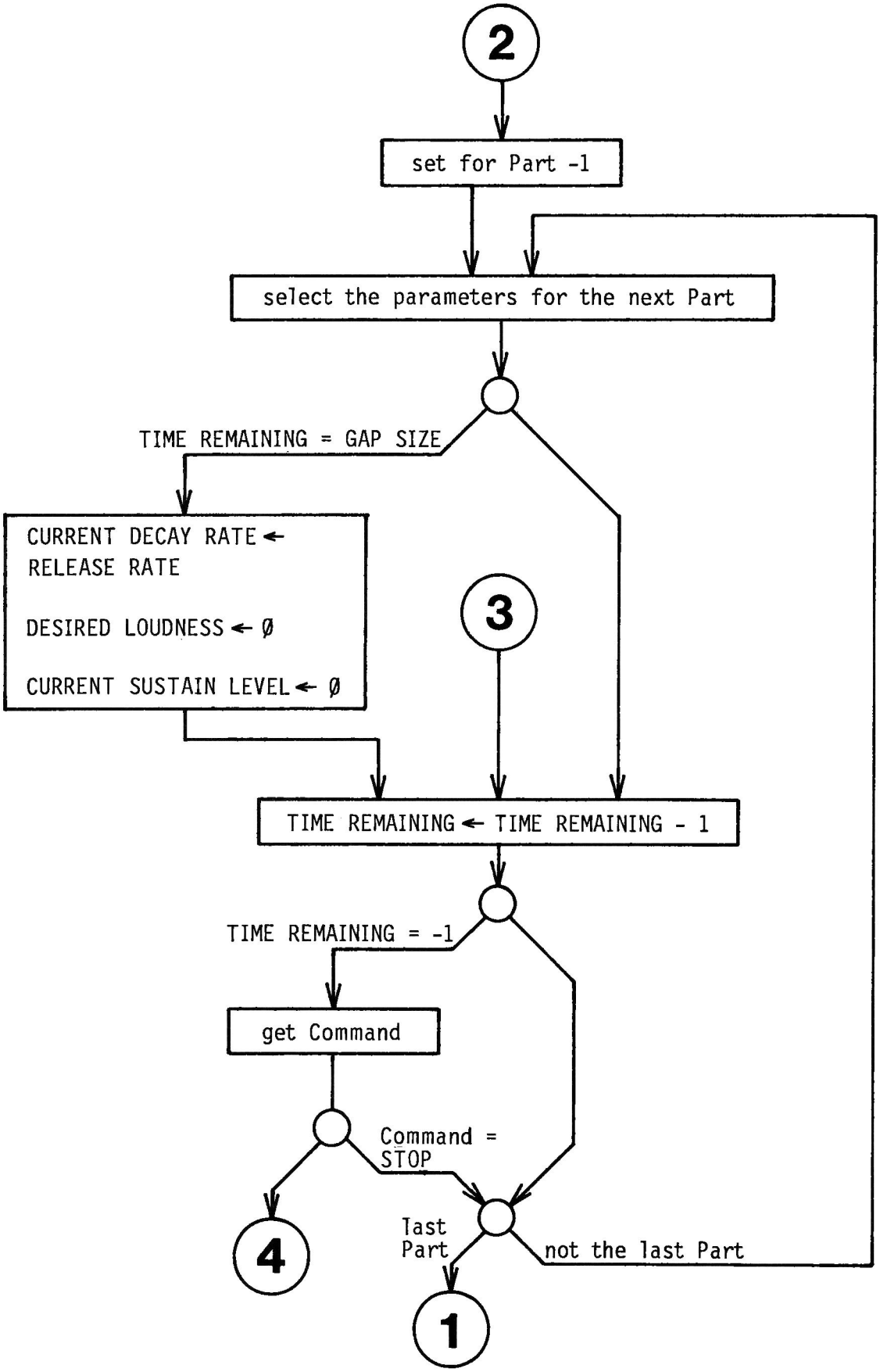
The "YALP" program (from the PERFORM sample session) can be modifed to use FLASH as follows:

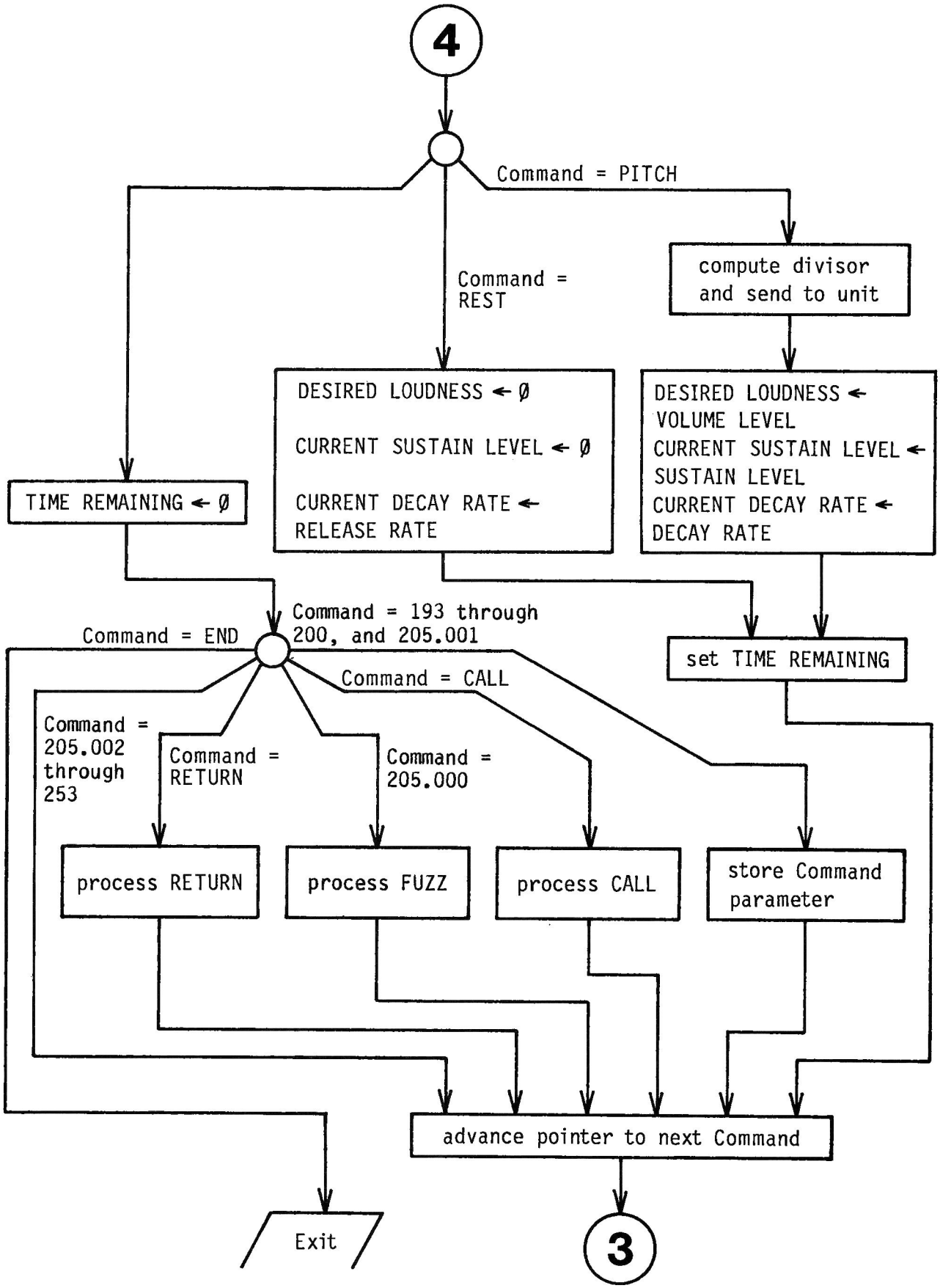
```
]FP
]LOAD YALP
]2Ø PRINT CHR$(4)"BLOAD FLASH"
]17Ø GR
]18Ø FOR A=1 TO PEEK(ADRS)
]19Ø COLOR=2 : HLIN Ø,39 AT A*4-2
]2ØØ COLOR=13 : PLOT 15,A*4-2
]21Ø NEXT
]22Ø CALL 36866 : TEXT : GOTO 7Ø
]SAVE NEW YALP
]RUN
SONG NAME?DIXIE BOOGIE
```

(song plays with display)

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
Command = END
Command = 193 through 200, and 205.001
set TIME REMAINING
Command = CALL
Command = 205.002 through 253
Command = RETURN
Command = 205.000
process RETURN
process FUZZ
process CALL
store Command parameter
advance pointer to next Command
Exit
3

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.

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 (on the MC16, the stereo position code is always Ø since position is indicated by slot number). 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 MC16 units (meaningless in songs that have more than six parts), and the least significant four bits indicate stereo for the MC1 or for three MC16's. In each 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 "initial 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. On the MC1, 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". FUZZ is not available on the MC16.

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 MC1 is limited to values from 3Ø to 174 (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 MC1 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

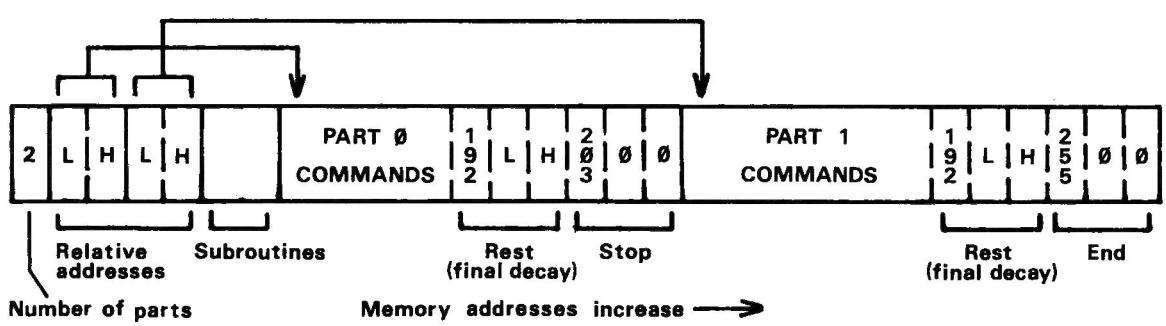
RELATIVE ADDRESSES

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



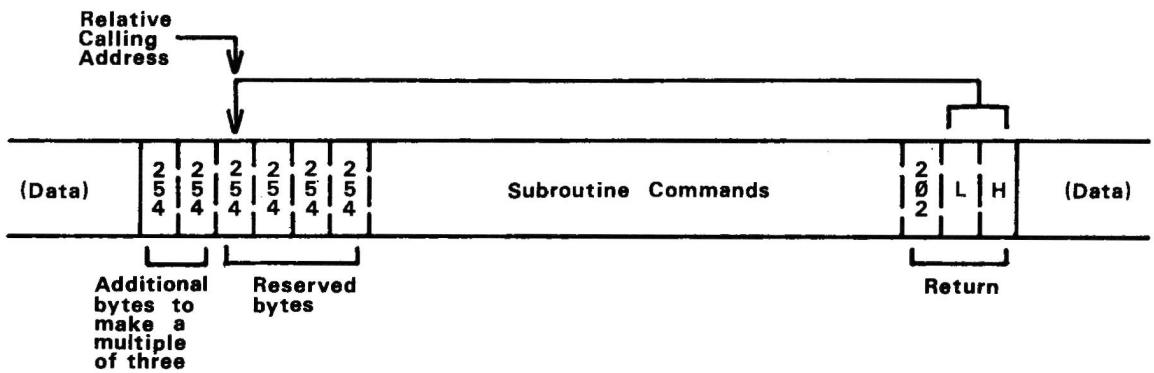
PART DATA

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 precede the first command of the subroutine. There must be two times as many reserved bytes as the number of parts. These reserved bytes must be preceded by at least 1 additional byte(s), and the number of additional bytes plus the number of reserved bytes must be evenly divisible by 3. See the diagram below.

Subroutine (in two part song data)



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.

RATE COMMAND

The RATE command is a rather unusual command. It is used to dynamically control playback speed for all parts. Like FUZZ, it is identified by the first two bytes. The third byte of the command indicates the new playback rate. "Rates" are related to "speeds" by the formula RATE=SPEED-11*(number of parts). During playback, the number of time periods per second is approximately 92,773/SPEED or 92,773/(RATE+11*(number of parts)). Previously, the TEMPO commmand (available only on the MC16) was used for playback tempo control. TEMPO commands can generally be replaced with RATE commands using the formula RATE=(TEMPO/19.17)-1-11*(number of parts).

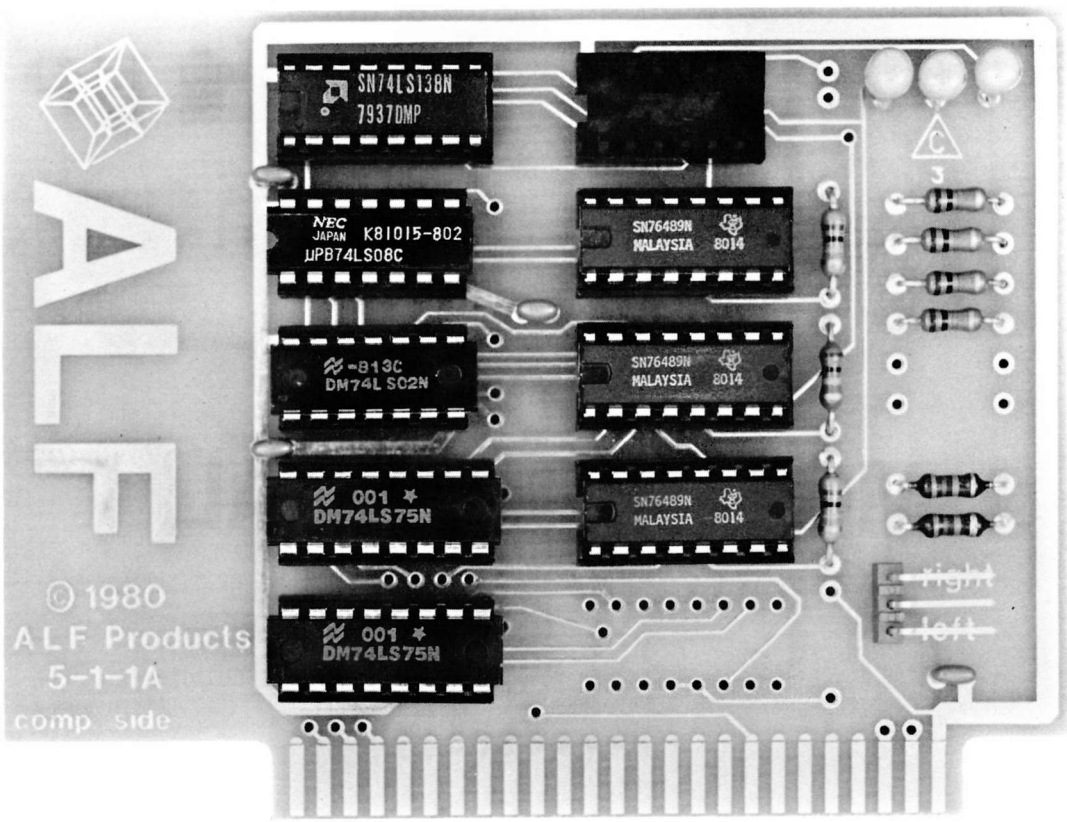
TEMPORARIES

PERFORM uses locations 6-C, DØ-D1, and DD-EF for storage of temporary values during execution. FLASH uses these locations, plus F9-FD.

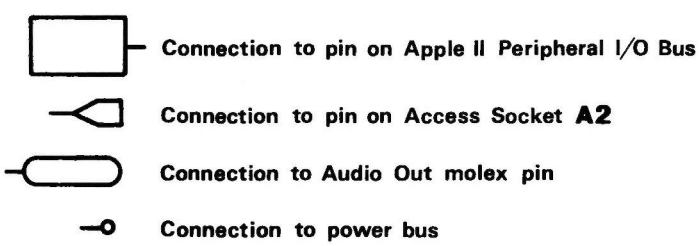
COMMAND NUMBERS

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 Ø1	2Ø5 ØØ1	RATE
CD **	2Ø5 ***	(where ** or *** are > 1) no operation
CE-FD	2Ø6-253	no operation
FE	254	precedes subroutines, treated as END if found
FF	255	END

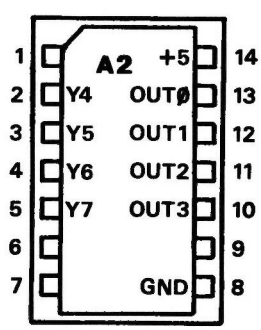
10
MC1 TECHNICAL

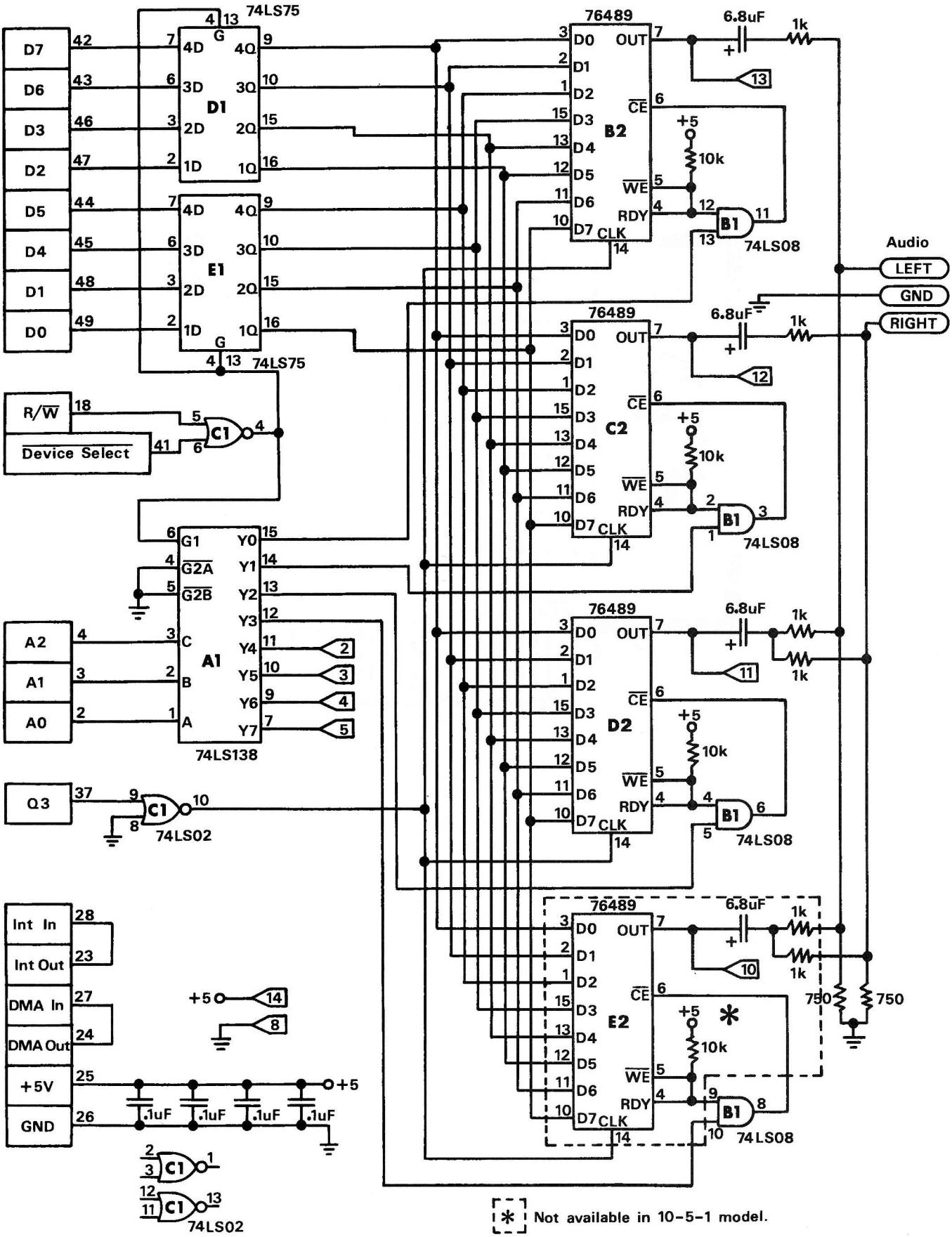


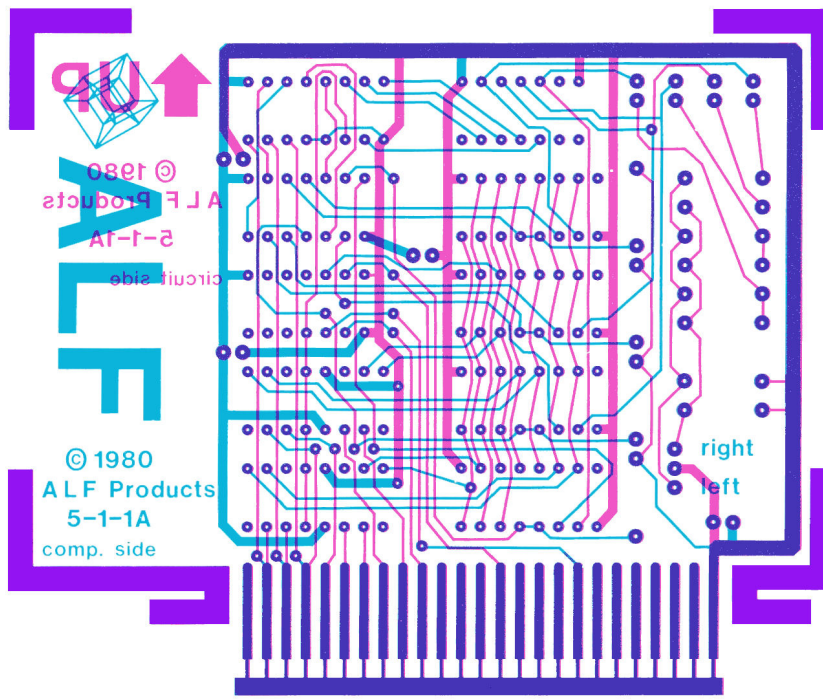
SCHEMATIC TERMINALS



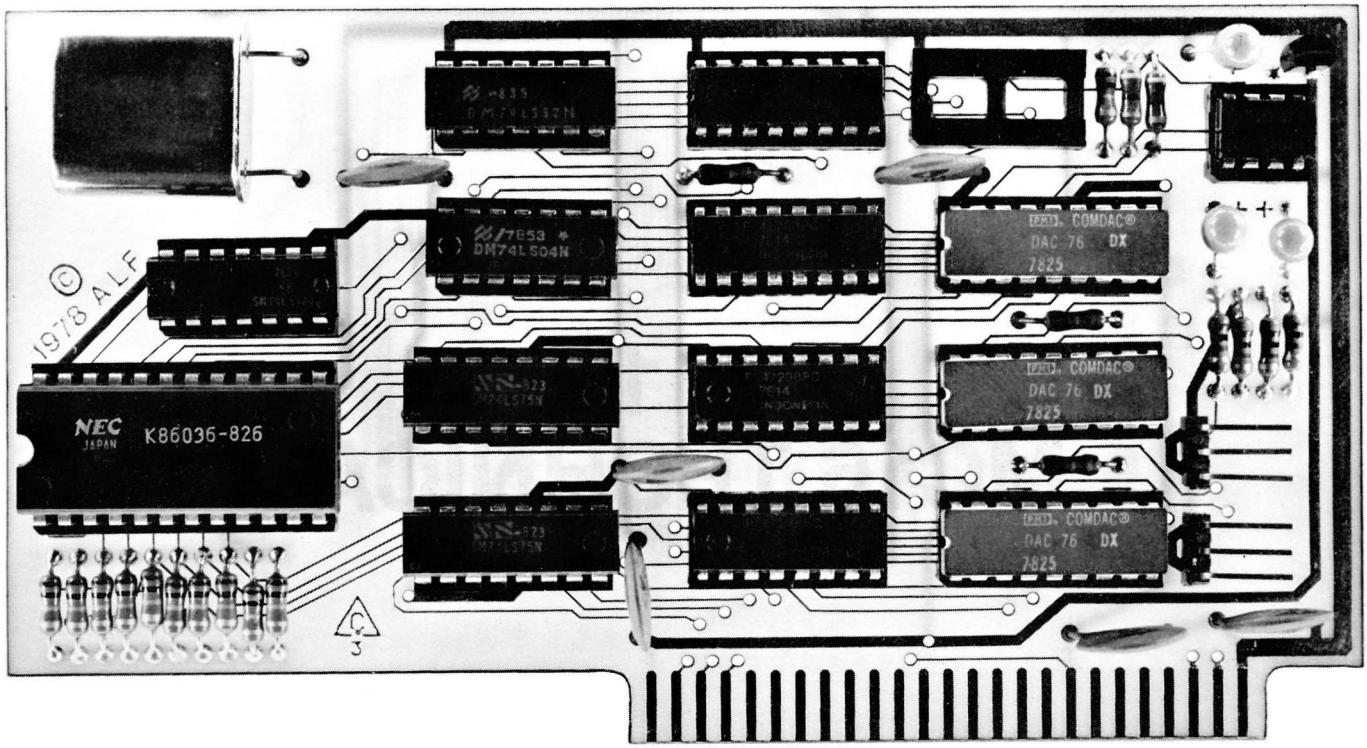
ACCESS SOCKET

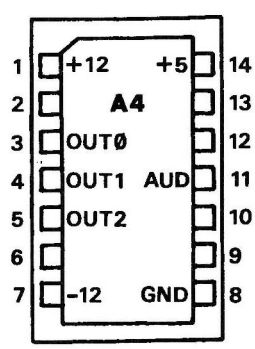



74LS75
D1
E1
D7
D6
D3
D2
D5
D4
D1
D0
R/W
Device Select
C1
74LS02
74LS138
A1
A2
A1
A0
Q3
Int In
Int Out
DMA In
DMA Out
+5V
GND
.1uF
76489
B2
C2
D2
E2
B1
74LS08
6.8uF
1k
10k
750
Audio
LEFT
GND
RIGHT
Not available in 10-5-1 model.


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

11
MC16 TECHNICAL





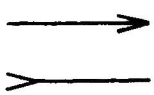
ACCESS SOCKET

AUDIO OUTPUTS

Impedance: 7ØØ ohms typical. Output: Ø.91 volts peak.

SCHEMATIC TERMINALS

Connection to pin on Apple II Peripheral I/O Bus

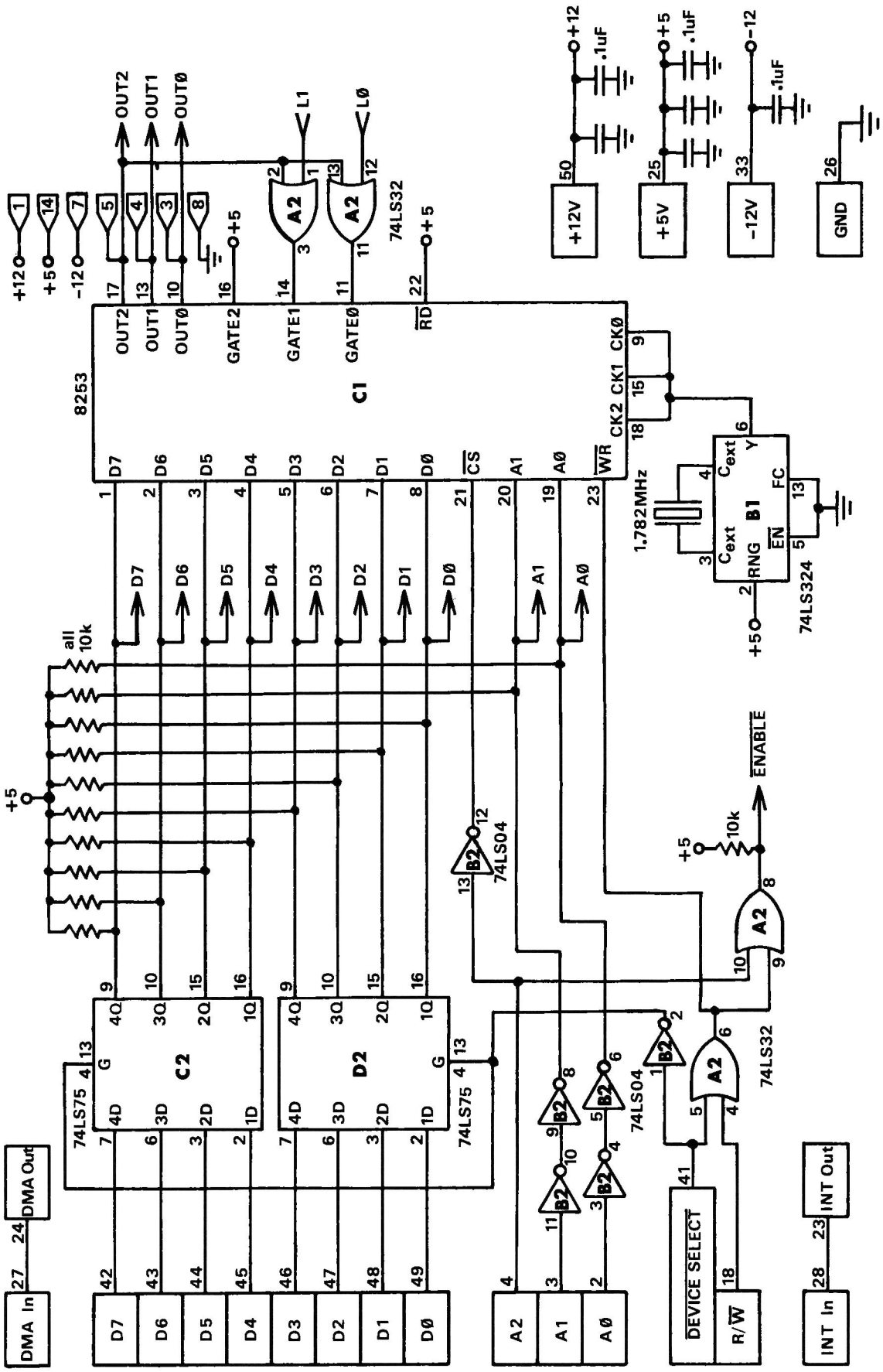


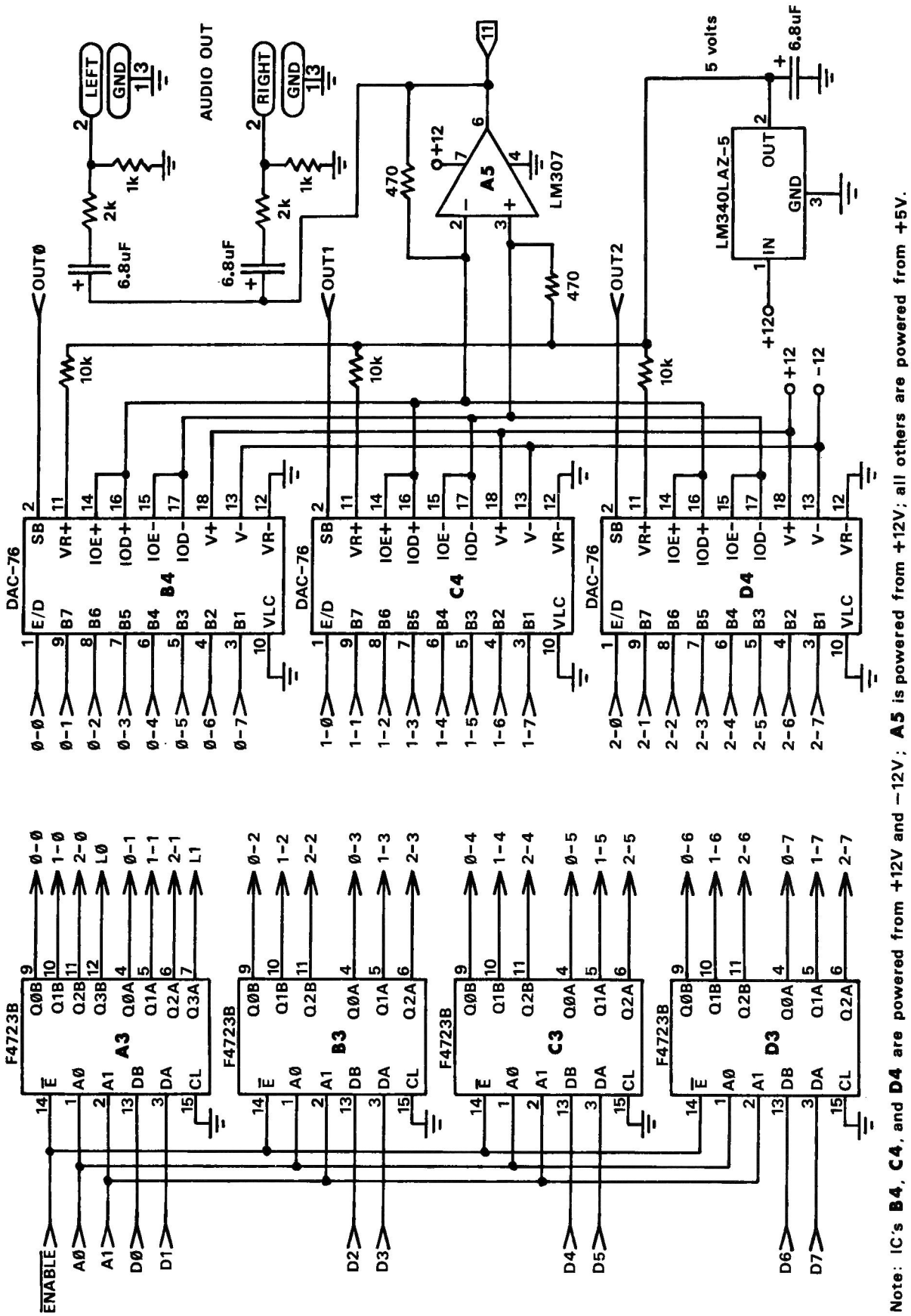
Internal connections

Connection to Audio Out molex pin

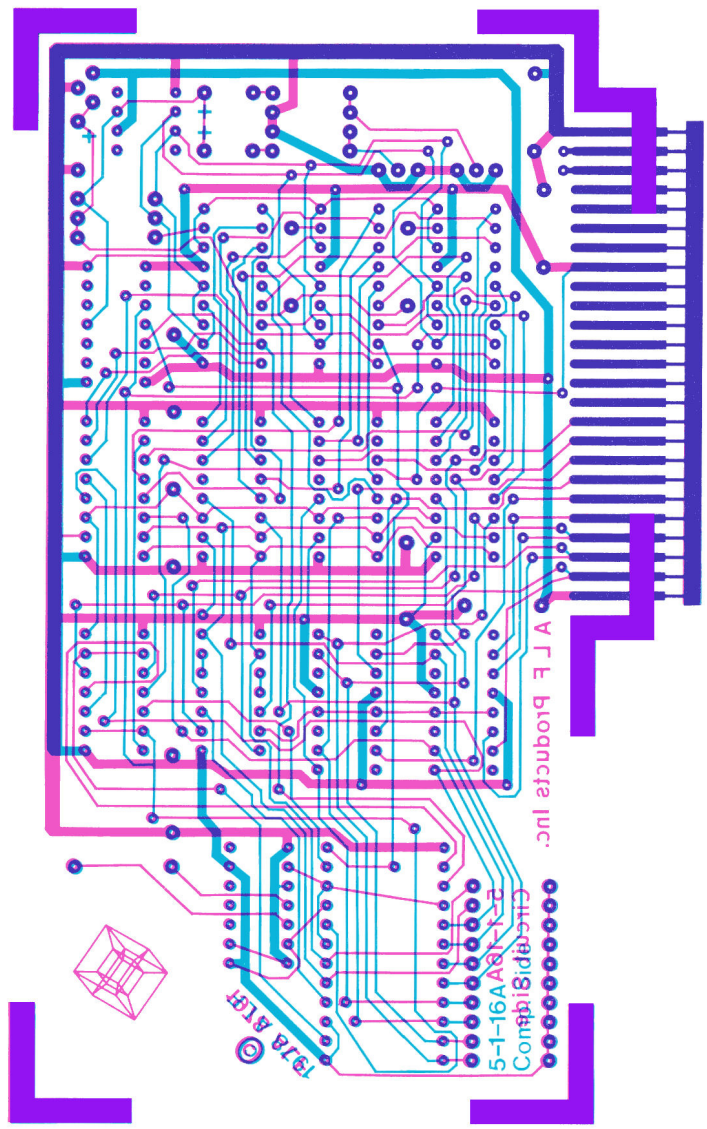
Connection to pin on Access Socket **A4**

Boldface characters on schematic (eg. **C2**) refer to component locations.
See silkscreen artwork for locations.


+12
+5
-12
OUT2
OUT1
OUTØ
L1
LØ
A2
74LS32
+5
+12
.1uF
+5
.1uF
-12
.1uF
+12V
+5V
-12V
GND
8253
C1
GATE2
GATE1
GATEØ
RD
CK2 CK1 CKØ
WR
CS
A1
AØ
D7
D6
D5
D4
D3
D2
D1
DØ
1.782MHz
B1
74LS324
all
10k
C2
D2
74LS75
74LS04
ENABLE
74LS32
DMA In
DMA Out
INT In
INT Out
DEVICE SELECT
R/W
A2
A1
AØ



Note: IC's **B4**, **C4**, and **D4** are powered from +12V and −12V; **A5** is powered from +12V; all others are powered from +5V.


5-1-16A
Circuit Side
ALF Products Inc.
1978 ALF

INDEX

. : 2-6, 2-8, 2-37
* : 2-3, 2-4, 2-34
***DISK: 2-32

10-1-2 Mono Cable: 1-1 to 1-2
6502 Programming: 7-2

Accidental Control Symbols: 2-30
ADSR Envelopes: 2-22 TO 2-23
Album Generation: 5-2
"America": 2-1 to 2-12
APPEND Command: 4-1, 4-4 to 4-5
Apple IIe & Apple III, Using ENTRY: 2-43
Applesoft: 2-42, 9-1
Asterisk: 2-3, 2-4, 2-34
ATTACK: 2-22, 2-23, 2-37, 9-7, 9-12, 9-15
Attempted Humor: 2-3, 2-5, 2-7, 2-8, 2-9, 2-16, 2-23, 2-24, 2-32
Audio Cable Connection: 1-1 to 1-2
AUXILIARY Command: 4-4

Backup: 1-5, 2-41
Beaming: 2-27
Binary File Conversion: 9-1, 9-4
BLOAD Command: 4-6
BSAVE Command: 4-6

CALL: 2-18, 2-38, 9-9, 9-10, 9-13, 9-14
CHANGE Command: 4-2
Channel Number Command: 9-10, 9-13, 9-15
Circuit Card Photo: MC1: 10-1, MC16: 11-1
Clicks: 2-6
Color of Playback Blocks: 2-31 to 2-32
Command Numbers: 9-15
Command Summary, ENTRY: 2-29 to 2-39
Command Summary, ENVELOPE: 3-3
Command Summary, MLIST: 4-12
Command Summary, PROCESS: 4-7
Composition, ENTRY: 2-1 to 2-27
Configuration, Song: 9-3 to 9-4
Control-S: 4-9
Converting a Song to a Binary File: 9-1, 9-4 to 9-5
Copying Songs Without ENTRY: 2-41 to 2-42
Copyright Restrictions: 1-5
Correcting Mistakes: 2-12 to 2-15
Current Decay: 2-24 to 2-27, 9-8 to 9-9, 9-12
Current Loudness: 2-23 to 2-27, 2-37, 2-38, 2-39, 9-7
Current Sustain: 2-24 to 2-26, 9-7 to 9-9, 9-12
Cursor: 2-2 to 2-8

DECAY: 2-22, 2-38, 9-7 to 9-9, 9-11, 9-12, 9-15
Default Settings in ENTRY: 2-33 to 2-34
DEL: 2-13, 2-30, 2-32
Delete a Note: 2-13
DELETE Command: 2-33, 4-1
Desired Loudness: 2-23 to 2-27, 2-39, 9-7 to 9-9
DISCO Program: 5-1 to 5-3
Disk Software: 1-2 to 1-3
Divisor Calculation: 7-2 to 7-3, 8-3 to 8-4
DOS Commands From MLIST: 4-9
DOS Commands From PROCESS: 4-3
Dot: 2-6, 2-8, 2-37
Duration: 2-29 to 2-30

"Echo" Effect: 2-27
EDIT: 2-9, 2-15, 2-33, 2-35
END Command: 9-11
END in DISCO: 5-2 to 5-3
End Marker: 2-4, 2-5, 2-30
Entering a Simple Song: 2-1 to 2-12
Entering Rests: 2-15 to 2-16
ENTRY Program: Section 2, 9-1 to 9-6
ENTRY, Summary of Commands: 2-29 to 2-39
ENTRY Without Paddles: 2-43
ENTRY2 Program: 2-43
ENVELOPE Program: Section 3
Envelopes: 2-21 to 2-27, 2-37 to 2-39, Section 3
Enveology: 2-23
Execute Files: 5-2 to 5-3

FLASH Program: 9-6
Flat: 2-30, 2-35, 2-37
Flying Saucer: 2-2
FP: 4-9, 5-1, 9-1
Free: 2-1, 2-4, 2-35
Frequency: 7-1 to 7-2, 8-2 to 8-3, 8-4
FUZZ: 2-34, 2-38, 9-12, 9-15
FWIDTH Command: 4-9

GAP: 2-4, 2-22, 2-38, 9-12, 9-13, 9-15
Good Things to Know: 2-40 to 2-42
GOTO: 2-31

Hoople: 2-24

Incremental Karma: 2-32
Infinite Repeats: 2-19
Initial Speed, Reading: 9-4
Initialization of Music Card: 6-1 to 6-3, 7-1, 8-1 to 8-2, 9-2 to 9-3
INS: 2-13, 2-30
Insert a Note: 2-5, 2-13, 2-30
Installation: 1-2 to 1-5
Integer/Applesoft Switch: 2-42
Integer Basic: 2-41 to 2-42

KEY: 2-2, 2-35 to 2-36, 2-42
Key Signature: 2-7, 2-37

Left and Right Movement Controls: 2-4, 2-5, 2-30
LENGTH: 2-31, 2-37
Line 10: 1-3, 2-1, 3-3, 5-1, 5-3
LIST Command: 4-8
LOAD Command: 2-19, 2-33, 4-1, 4-8, 5-1
Loading and Saving Songs: 2-19 to 2-20
Loudness, Definition: 2-21

Measure: 2-6, 2-31, 2-36, 2-37
MEASURE: 2-31
Memory Requirements: 2-1
Menu Commands: 2-29 to 2-30
Menu Paddle: 2-3
Mistakes, How to Fix: 2-12 to 2-15
MLIST Program: 4-8 to 4-12
Movement Controls, Left and Right: 2-4, 2-5, 2-30
Music Card Hardware: MC1: Section 10, MC16: Section 11
Music Card Initialization: 6-1 to 6-3, 7-1, 8-1 to 8-2, 9-2 to 9-3
Music Notation: 2-27

Natural: 2-5, 2-30, 2-36, 2-37
NEW: 2-2, 2-15, 2-22, 2-33 to 2-34
New Part: 2-9
New Song: 2-2
Note Duration Menu Symbols: 2-29 to 2-30
Note Paddle: 2-2, 2-3, 2-37
Notes, Nonstandard Length: 2-20 to 2-21
Notetrinos: 2-24

Operating Tips: 1-4

Paddle 0: 2-3
Paddle 1: 2-2, 2-3, 2-37
Paddle Settings: 2-40
Paddles: 2-3
Paramatron, High Powered: 2-24
PART: 2-31
Part, Adding New: 2-9
Part Data: 9-13
Part, Definition: 2-2
Partial Starting Measure: 2-40
Percussion: 2-38 see also FUZZ
PERFORM: Section 9
PERFORM Block Diagram: 9-7 to 9-9
PERFORM Technical Description: 9-10 to 9-15
PERFORM Temporaries: 9-15
PITCH Commands: 9-12 to 9-13, 9-15
PLAY Command: 2-8, 2-10, 2-31, 5-1
PLAY Program: 5-1
Play Song Using BASIC, How to: 9-1 to 9-6
POKE: 2-38, 4-6, 4-12
PR# Command: 4-3, 4-9
Problem Checklist: 1-4 to 1-5
PROCESS Program: 4-1 to 4-7

QUARTER: 2-3, 2-20, 2-23, 2-36 , 2-37, 2-40, 2-41

RATE Command: 4-6, 4-12, 9-14
Recursive Subroutine: 2-19
Relative Addresses: 9-13
RELEASE: 2-22, 2-39, 9-12, 9-15
Repair: 1-4
Repair Diagram: MC1: 10-3, MC16: 11-5
Repeats: 2-16 to 2-19
Reset Button: 2-19, 2-35, 2-42, 5-1
REST: 2-15 to 2-16, 2-37, 9-13, 9-15
Rests at the Ends of Parts: 2-40
RETURN Command: 9-11, 9-14, 9-15
Right Movement: 2-4, 2-30
Rounds: 2-16 to 2-19
"Row, Row, Row Your Boat": 2-16 to 2-19

Sample Listing, MLIST: 4-10
Sample Session, PERFORM: 9-4 to 9-5
SAVE Command: 2-19, 2-32, 4-1, 4-5
Saving Songs: 2-19 to 2-20
Schematics: MC1: 10-1 to 10-2, MC16: 11-2 to 11-4
Sharp: 2-30, 2-35, 2-37
SLOT: 1-3, 2-1, 9-1
"Song" as only word for musical work: 2-15 to 2-16
Song Breakdown: 2-28
Song Configuration: 9-3
Song Data: 2-42, 9-13 to 9-14
SPEED: 2-15, 2-23, 2-34
START in DISCO: 5-2
STATUS Command: 4-2
Stereo: 1-1, 2-33, 2-34, 7-1, 9-10
STEREO: 2-15, 2-34, 2-38
STOP: 5-1, 9-8, 9-11, 9-13, 9-15
SUBROUTINE: 2-17, 2-35
Subroutine Data: 9-14
Subroutines: 2-16 to 2-19, 2-35, 2-38, 9-10 to 9-11
Summary of Commands: See Command Summary
SUSTAIN: 2-22, 2-39, 9-12, 9-13, 9-15

TEMPO: 2-39
Tempo Adjustment: 2-20 to 2-21
Temporaries, PERFORM: 9-15
TIE: 2-14, 2-37
TIME: 2-36
Time Periods: 2-40
Time Remaining: 9-9
Tips for Operation: 1-4
Tips on ENTRY: 2-40 to 2-42
TRANSPOSE: 2-18, 2-39, 2-41, 9-11, 9-15
Triplet: 2-17, 2-29 to 2-30
Troubleshooting: 1-4 to 1-5
Tuning Accuracy: 7-3, 8-4

Useless Questions: 2-16

VOLUME: 2-22, 2-39, 9-12, 9-13, 9-15

White Noise: 2-38
WIDTH Command: 4-3, 4-9

X-notes 2-21

YALP Program: 9-4 to 9-6