H A 8 - 2 B o a r d
a n d
M u s i c S o f t w a r e
D o c u m e n t a t i o n

Table of Contents

0.0) Introduction

The HA8-2 is a dual channel D/A (digital to analog) converter circuit with two 3300 hertz, 6 pole, low pass active filters. While the HA8-2 is designed for music synthesis, it can be used in virtually any D/A application including voice synthesis. The music software supplied with the HA8-2 can synthesize four parts of harmony (musical voices) simultaneously in stereo.

There is also an independent output for each channel which bypasses the active filters so that the analog outputs may be used for other purposes without frequency limitation.

1.0) Programming the HA8-2 without music software

If the HA8-2 is going to be used with the music software, this section can be skipped. The D/A converters on the HA8-2 respond to a memory write at an address which can be set by DIP switches on the board. Refer to section 2.1 "Setting the HA8-2 address" if it is necessary to change the memory address. In the following discussion, absolute voltage values are used, but in actuality, voltages may deviate slightly from board to board. The D/A converters on the HA8-2 board convert a binary number into a voltage between 0 and 5 volts. Each bit of a byte written to one of the D/A converters has a voltage associated with it. The high order bit (bit 7) has a voltage value of 2.5 volts. Each lower bit has a value of 1/2 of the previous bit. Bit 6, therefore, has a value of 1/2 of bit 7, or 1.25 volts. A table of bits and their corresponding voltage values follows:

	Bit	Voltage Value
(high order)	7	2.5
	6	1.25
	5	0.625
	4	0.3125
	3	0.15625
	2	0.078125
	1	0.0390625
(low order)	0	0.01953125

When a number is written to one of the D/A converters, the voltage values of the bits which are ON are summed. For example, to obtain a voltage of 3.75 volts, a byte containing a 300 octal must be written to the D/A converter (300 octal has bits 7 and 6 ON).

There are two outputs available from each of the D/A converters. The outputs labeled DAC 0 and DAC 1 are the buffered outputs of the D/A converters. Scaling of the output voltages should be done to correct for errors in the 12 volt power supply. The other outputs are located in the top center of the board and are labeled CNL 0 and CNL 1. These outputs are filtered by independent 3300 hertz, 6 pole, low pass active filters for each channel. These filters attenuate the switching noise caused by the transition of the D/A converters from one voltage to another. The output peak to peak voltage for CNL 0 and CNL 1 can be controlled by the two trim pots, one for each channel, up to a maximum of 1 volt. The output from the filtered outputs is AC coupled. The minimum load impedance is 10k ohms.

2.0) Configuring the HA8-2

Before the HA8-2 can be used, its address and peak to peak output voltages must be set. This is described in detail below.

2.1) Setting the HA8-2's address

The HA8-2 responds only to memory writes to an adjacent even/odd pair of addresses which can be selected by the DIP switches. The music software supplied with the HA8-2 uses addresses 0 and 1. Address 0 corresponds to all address switches in the open (off) position and the board enable (bottom most) switch in the closed (on) position. If the music software supplied with the system is going to be used, set the DIP switches in this configuration. (The board is shipped with the DIP switches in this position.) Address bits 15 through 1 are compared with the DIP switch settings. Bit 15 is compared with the top-most switch of the upper DIP switch (S2), bit 14 with the next lower switch, etc. A table of address bits and their corresponding switches follows:

Address Bit	Switch	Pole
15	Upper(S2)	8
14	Upper "	7
13	Upper "	6
12	Upper "	5
11	Upper "	4
10	Upper "	3
9	Upper "	2
8	Upper "	1
7	Lower(S1)	8
6	Lower "	7
5	Lower "	6
4	Lower "	5
3	Lower "	4
2	Lower "	3
1	Lower "	2
Board Enable	Lower "	1

Bit 0 is not set because the board uses it internally. When an address bit and the corresponding DIP switch are both in the same state (i.e., either both ON or both OFF), they are equal. When board enable is on and address bits 15 through 1 are all equal to their corresponding DIP switches, and a memory write is also occurring, then one of the two D/A converters will respond to the memory write. Address bit 0 is used to select one of the two D/A converters. If address bit 0 is OFF, then D/A channel 0 (outputs labeled CNL 0 and DAC 0) is selected and will respond to the memory write, otherwise D/A channel 1 (outputs labeled CNL 1 and DAC 1) is selected. Note that the HA8-2 and memory can both

respond to the same address. When this occurs the data byte is written to both memory and the HA8-2. However the memory can be read without affecting the HA8-2 because the HA8-2 only responds to memory writes. This provides a method of reading out the digital value being converted to an analog voltage.

2.2) Setting the peak to peak output voltage

When using the D/A converters to drive external circuits such as a phono amplifier, caution must be exercised not to over-drive the external circuit. The following steps will help insure that the filtered output of the D/A will not over-drive an external circuit. With the HA8-2 board installed in the backplane, rotate both trim pots to the extreme clockwise position. This reduces the output to minimum. Connect the HA8-2 audio connectors to the external amplifier and set the volume control on the external amplifier to a normal listening level. Refer to the section 5.0 "The PLAY program", and play one of the demo songs. While the song is playing (the front panel display will be off during this time), slowly rotate the trim pots counter-clockwise until the volume level is approximately at a normal listening level. Some sound should be heard before rotating the pots more than a quarter of a turn. If no sound is heard verify the following:

1) The DIP switches are set in the proper position.

2) The amplifier has the correct input selected. For example if the HA8-2 is connected to the "PHONO" input of the amplifier, then the amplifier should have the "PHONO" input selected.

3) The speakers are connected to the amplifier.

4) The amplifier has power and is turned on.

If, after verifying that the amplifier is good, the board will not function, refer to section 7.2 "Repair and warranty".

3.0) Music software overview

The music software supplied with the HA8-2 provides an easy to use interface between the sheet music for a song and the D/A converters on the HA8-2 board. A song file is created with the text editor from the sheet music. The song file usually has an extension of "MUS". This file is then compiled into a playable form using the program "COMPOSE". At compilation time the following information is specified:

1) The name of the song file.

2) The name of the playable file. (This is the output of the compiler.)

3) The number of voices (parts of harmony).

4) The waveform information for each voice. (The waveform controls the timbre of the sound.)

5) Whether or not duplicate pitches on the same channel are to be suppressed.

6) The speed at which the song is to be played.

The playable file usually has an extension of "PLA". The output of the compiler can be played at any time using the "PLAY" program. The song file needs recompiling only if one of the above six items requires changing.

4.0) The "COMPOSE" compiler

The COMPOSE program compiles the song file into a form which the PLAY program can use. To run the COMPOSE program enter the following operating system command:

```
RUN dev:COMPOSE
```

"dev" is "SYO" if the COMPOSE compiler is on SYO
or "SY1" if the COMPOSE compiler is on SY1.

The COMPOSE program requires 24K of memory to run. It may require more memory for very large song files. When COMPOSE starts to execute, it will print a series of questions. Each question is explained below.

4.1) "Input filename: "

COMPOSE is requesting the name of the song file. If the device is not specified, it will default to the same device specified on the RUN command. If the extension is not specified, it will default to "MUS". If an error message is printed, enter the filename of the song file in the correct format.

4.2) "Output filename: "

COMPOSE is requesting the name of the output file. This will be the file which the PLAY program uses. If the device is not specified, it will default to the same device as the input file. If the extension is not specified, it will default to "PLA". If an error message is printed enter the filename in the correct format.

4.3) "How many voices?: "

COMPOSE is requesting the number of voices in the song file. The valid responses are "1", "2", "3", or "4". If any other value is entered, COMPOSE will respond with the error message "?The number of voices must be an integer from 1 to 4".

4.4) Entering the waveform information for each voice

If a note of a given pitch (frequency) is played successively on two different musical instruments with the same loudness, a listener can distinguish between the two sounds. Each instrument has its own musical quality (timbre). The difference in timbre is due to different instruments generating different harmonics (multiples) of the fundamental frequency in addition to the fundamental frequency. Each harmonic has an amplitude and

phase shift values relative to the fundamental. The fundamental frequency and its harmonics form the characteristic waveform for the pitch and loudness of an instrument. Although the phase and amplitude relationships of the harmonics change with pitch and loudness, this music software permits only one waveform per voice. For each voice, COMPOSE will request the amplitude and phase relationships of the fundamental and higher harmonics. COMPOSE can generate a waveform table with up to nine harmonic components. Unfortunately, there is no good way to guess the harmonic components of a waveform from hearing it or from a graph of the waveform. If a graph or picture of a waveform is available, the WAVE program can be used to give good approximations of the harmonic components (Refer to section 6.0 "The WAVE program" for more information). If a graph or picture of a waveform is not available, then it will be necessary to experiment in order to get the timbre desired. After listening to a musical instrument a rough determination of the harmonic components may be made by using the following simple rules: The fundamental gives the pitch of the note. The second harmonic adds clearness and brilliance. The third harmonic adds brilliance and contributes a hollow, throaty or nasal quality. The fourth harmonic adds more brilliance and even shrillness. The fifth harmonic adds a rich, somewhat horn-like quality. The sixth harmonic adds a delicate shrillness of nasal quality. The seventh and ninth harmonics introduce dissonance and roughness and harshness. The higher the relative amplitude of a harmonic the more the qualities of that harmonic contribute to the sound. While there is evidence to suggest that phase shift does contribute to the attack of a note, the ear is not very sensitive to these shifts. Hence, it is by far much more important to get the correct amplitudes for the harmonics rather than the phase shifts.

A waveform with the following harmonic components will produce a pleasing timbre:

	Harmonic	Amplitude	Phase
(fundamental)	1	2	0
	2	1	90
	3	1	0

Each harmonic component specified requires about one minute of computation time. If the desired waveform already exists in a previously compiled song, it can be copied rather than recomputing it. Copying waveforms greatly reduces the compilation time.

4.4.1) "How many harmonics in waveform for voice 'n'?: "

COMPOSE is requesting the number of harmonic components which comprise the waveform for voice 'n'. The valid responses are "0", "1", "2", "3", "4", "5", "6", "7", "8", or "9". A response of "0" means that no harmonics are to be computed, and COMPOSE assumes that the waveform is to be copied from a previously compiled song. In this case, COMPOSE issues the informatory message "[Waveform will be copied from disk.]". If an integer from "1" to "9" is entered, then COMPOSE will request the amplitude and phase shift information for each harmonic. If an invalid number is entered, then COMPOSE will issue the error message "?The number of harmonics must be an integer from 1 to 9.". Following a prompt from COMPOSE a valid number can then be entered.

4.4.2) Amplitude and phase shift values

For each harmonic in the waveform of a voice, COMPOSE will request the amplitude and phase shift values. The phase shift for the harmonics are entered in degrees. The amplitude values for the harmonics must be positive. Only the ratio of the amplitudes to one another is important. For example, if in the case of harmonics with amplitudes 0.6 for harmonic 1 and 0.4 for harmonic 2, it makes no difference if the amplitudes for harmonics 1 and 2 are entered as 60 and 40, 6 and 4, or 3 and 2.

4.4.2.1) "What is the amplitude for harmonic 'n'?: "

COMPOSE is requesting the relative amplitude information for a harmonic component of a voice. Enter the value as described in the section above. If the number is not positive, COMPOSE will issue the error message "?Amplitude must not be negative.". If this error message occurs, a prompt from compose will be typed and the correct value can be entered.

4.4.2.2) "What is the phase shift (in degrees) for harmonic 'n'?: "

COMPOSE is requesting the phase shift information for a harmonic component of a voice. The phase shift value entered must be in degrees.

4.5) Copying waveforms

For each voice which had "0" entered for the number of harmonic components, COMPOSE will copy the waveform from a previously compiled song on disk. For each voice compose will prompt for the filename of the song and the voice within that

song which is to be copied.

4.5.1) "Which file has compiled waveform table for voice 'n'?: "

COMPOSE is requesting the filename for the previously compiled song which has the waveform to be copied for voice 'n'. If the device is not specified, then the same device as the output file is used. The default extension is "PLA". If any error messages occur, the filename can be entered following the prompt from COMPOSE.

4.5.2) "Which voice is to be copied?: "

Since there are up to four voices compiled into a song, COMPOSE is requesting the voice within the file to be copied. The valid responses are "1", "2", "3", or "4". If the response is not valid, COMPOSE will issue the error message "?The number of voices must be an integer from 1 to 4.". The correct value can then be entered following the prompt from COMPOSE.

4.6) "Are duplicate pitches on the same channel to be suppressed? (Yes): "

When the same pitch (frequency) is synthesized for two voices on the same D/A channel, cancellation may occur. This is caused by the phase relationship between the two voices. If the voices are 180 degrees out of phase, then total cancellation occurs and no sound occurs. On the other hand, if the voices are in phase, then the loudness of the pitch is intensified. In natural music it is virtually impossible for two instruments to cancel perfectly. But for digitally synthesized music, the frequency of the two voices is precisely the same. Therefore cancellation does occur. To avoid this COMPOSE will suppress one pitch and permit the other. By suppressing one of the duplicate pitches they cannot cancel each other out. Respond "Yes" if it is desired to have duplicate pitch suppression; otherwise, respond "No". The default is "Yes". If an incorrect response is entered, COMPOSE will issue the message "Respond Yes or No.". COMPOSE will then prompt for corrected input.

4.7) The song tempo

The speed at which a song is played is controlled by two factors: which note duration sets the beat, and the number of beats per minute. The response to the following questions will determine the song tempo.

4.7.1) "Which note sets the beat?: "

Enter the note duration which sets the beat from the following table:

Beat Note	Response
Whole	1
Half	2
Quarter	4
Eighth	8
Sixteenth	16
Thirty-second	32
Sixty-fourth	64

For example, in common time the quarter note sets the beat. Therefore the correct response to the above question is "4". If an invalid response is entered COMPOSE will issue the following error message "?The value must be 1, 2, 4, 8, 16, 32, or 64.". The correct value can be entered following the prompt.

4.7.2) "How many beats per minute?: "

This number must be a positive number and is the same as the metronome setting. The higher the number, the faster the song is played. If a positive number was not entered, COMPOSE will issue the error message "?The value must be positive.". COMPOSE will then prompt for the value again.

4.8) Other keying errors

When an invalid character is keyed in a numeric value, COMPOSE will issue the error message "?Number is not valid.". COMPOSE will then prompt for the value again.

5.0) The PLAY program

The PLAY program plays the output from COMPOSE. To load PLAY into memory, enter the following operating system command:

```
dev:PLAY

where "dev" is "SYO" if PLAY is stored on SYO:
            or "SY1" if PLAY is stored on SY1:
```

The songs which are to be played are specified in a "request string". A request string is a list of file specifications separated by commas. Each file specification is written in the normal format of "dev:filename.PLA". For example, to play the songs SY1:F001.PLA and SY1:F002.PLA, the request string would be:

```
SY1:F001.PLA,SY1:F002.PLA
```

PLAY accepts the request string in one of three places: on the command line, as console input, or in a control file on disk.

Command line format

When the request string is placed on the command line, the operating system command which executes PLAY is as follows:

```
SY1:PLAY SY1:F001.PLA,SY1:F002.PLA
```

Console input format

If there is no request string on the command line, PLAY will prompt for console input with a "*". For example:

```
SY1:PLAY
*SY1:F001.PLA,SY1:F002.PLA
```

Control files

PLAY has the ability to process request strings which are on disk. The name of the control file is specified in a request string with an "@" prefixing the file specification. For example, suppose that the control file SY1:F00.CTL has the following request string:

```
SY1:F001.PLA,SY1:F002.PLA
```

Then the songs SY1:F001.PLA and SY1:F002.PLA could be played in either of the following two ways:

```
SY1:PLAY @SY1:FOO.CTL
```

or

```
SY1:PLAY
*@SY1:FOO.CTL
```

Once a control file specification is encountered in a request string, the rest of the request string is ignored. A request string in a control file may also contain a control file specification. For example, suppose that SY1:FOO.CTL contains:

```
SY1:FOO1.PLA,SY1:FOO2.PLA,@SY1:FOO.CTL
```

Then the command:

```
SY1:PLAY @SY1:FOO.CTL
```

will cause the songs SY1:FOO1.PLA and SY1:FOO2.PLA to continue to play over and over.

File specification defaults

If an extension is not given for a song to be played, then PLAY will default to the same extension as the last song played. If this is the first song played, then the default is PLA. If a device is not specified for a song to be played, then PLAY will default to the same device as the last song played. If this is the first song played, then the default device is the same device used to load PLAY.

If an extension is not given for a control file, then PLAY will default to the same extension as the last control file used. If this is the first control file used, then the default is CTL. If a device is not specified for a control file, then PLAY will default to the same device as the last control file used. If this is the first control file used, then the default device is the same device used to load PLAY.

Interrupting PLAY

The keys Control-A, Control-B, and Control-C are used to control PLAY while it is playing a song. DO NOT use Control-Z to stop the execution of PLAY.

If a song is not actually playing, then all of the control keys cause PLAY to stop execution. If PLAY is playing a song then

the control keys are defined as follows:

Control-A

When a Control-A is struck on the console, the current song stops playing and PLAY will immediately terminate execution. Use Control-A to stop the execution of PLAY, not Control-Z.

Control-B

When a Control-B is struck on the console, the current song stops playing, the rest of the current request string is ignored, and a new request string is prompted for from the console.

Control-C

When a Control-C is struck on the console, the current song stops playing, and the next file specification in the request string is processed.

6.0) The WAVE program

WAVE is a BASIC program which performs a discrete Fourier transformation on a waveform. The results of the transformation are the amplitude and phase shift components of the waveform. The harmonic components can then be used as input to COMPOSE. To execute the WAVE program, issue the following commands:

```
RUN dev:BASIC

"dev" is "SY0" if BASIC is on SY0
      or "SY1" if BASIC is on SY1.
```

Once BASIC is loaded, a prompt of an "*" will be typed on the console. Then the following BASIC command should be entered:

```
OLD "dev:WAVE.BAS"

(The quotes on the above command are required by BASIC)

"dev" is "SY0" if WAVE is on SY0
      or "SY1" if WAVE is on SY1.
```

When the WAVE program is loaded it can be executed it by issuing the BASIC command "RUN".

6.1) Preparation of the waveform data

The preparation of the waveform data depends on the number of harmonic components in the waveform. In order to accurately analyze the waveform, all of its significant harmonic components must be considered even though some of those components may be too high in frequency to pass through the 3300 hertz filters. For instance, suppose that a waveform has five significant harmonic components and the highest pitch played using the waveform is "A6". Since the fundamental frequency of "A6" is 880 hertz, the harmonics are:

	Harmonic	Frequency (hertz)
(fundamental)	1	880
	2	1760
	3	2640
	4	3520
	5	4400

The fourth and fifth harmonics will be attenuated by the filters. Therefore only the first three harmonics should be computed, but all five significant harmonics must be considered in order for the first three to be computed correctly.

See appendix A for a table of pitches and the highest

harmonic component which will pass through the filters. The highest harmonic which will pass through the filters should be used as an upper limit of the number of harmonics computed. The highest significant harmonic component to be considered will have to be guessed. The contribution of harmonics to the timbre are discussed in section 4.4. These guidelines can be used to determine if the harmonic is present. If they are present, WAVE will compute the amount they contribute to the waveform. In general it is better to guess too high rather than too low. The only drawback to guessing too high is the number of intervals that the waveform will have to be broken into will increase. The number of harmonics considered will always have to be at least as large as the number of harmonics computed.

Once the number of harmonics to be considered is determined, one complete cycle of the waveform will be divided into twice that number plus two equally spaced intervals. For example, if three harmonics are considered, then one complete cycle of the waveform will be divided into eight equally spaced intervals. The program will request the instantaneous value of the waveform at each of these intervals.

6.2) Responding to WAVE

Once WAVE begins it will request the following:

1) The number of harmonics to be considered.

2) The number of harmonics to be computed.

3) The instantaneous value at each of the intervals.

WAVE will determine the number of intervals required and the amplitude and phase shift for each harmonic computed. The amplitudes will be normalized so that the sum of the amplitudes is 1.

6.2.1) "How many harmonics are to be considered?: "

This must be an integer at least as large as the number of harmonics computed. This number should be the number of harmonic components determined to be in the waveform -- even if they are too high in frequency to be used by the music software. This will insure that the computed (useable) harmonics are as accurate as possible. If the number entered is not a positive integer, WAVE will issue the error message "?The number of harmonics considered must be a positive integer.". WAVE will then prompt again and the correct response can be entered.

6.2.2) "How many harmonics are to be computed?: "

This must be a positive integer no larger than the number entered in response to 6.2.1 above. This number specifies the highest harmonic useable by the music software. Harmonics above 3300 hertz are not useable because they will be attenuated by the filters. If the number entered is not a positive integer less than or equal to the number from 6.2.1, WAVE will issue the error message "?The number of harmonics computed must be a positive integer no greater than 'n'.". 'n' is the number specified in response to 6.2.1. When WAVE prompts for the number, the correct response can be entered.

6.2.3) "Divide the waveform into 'n' equally spaced intervals."

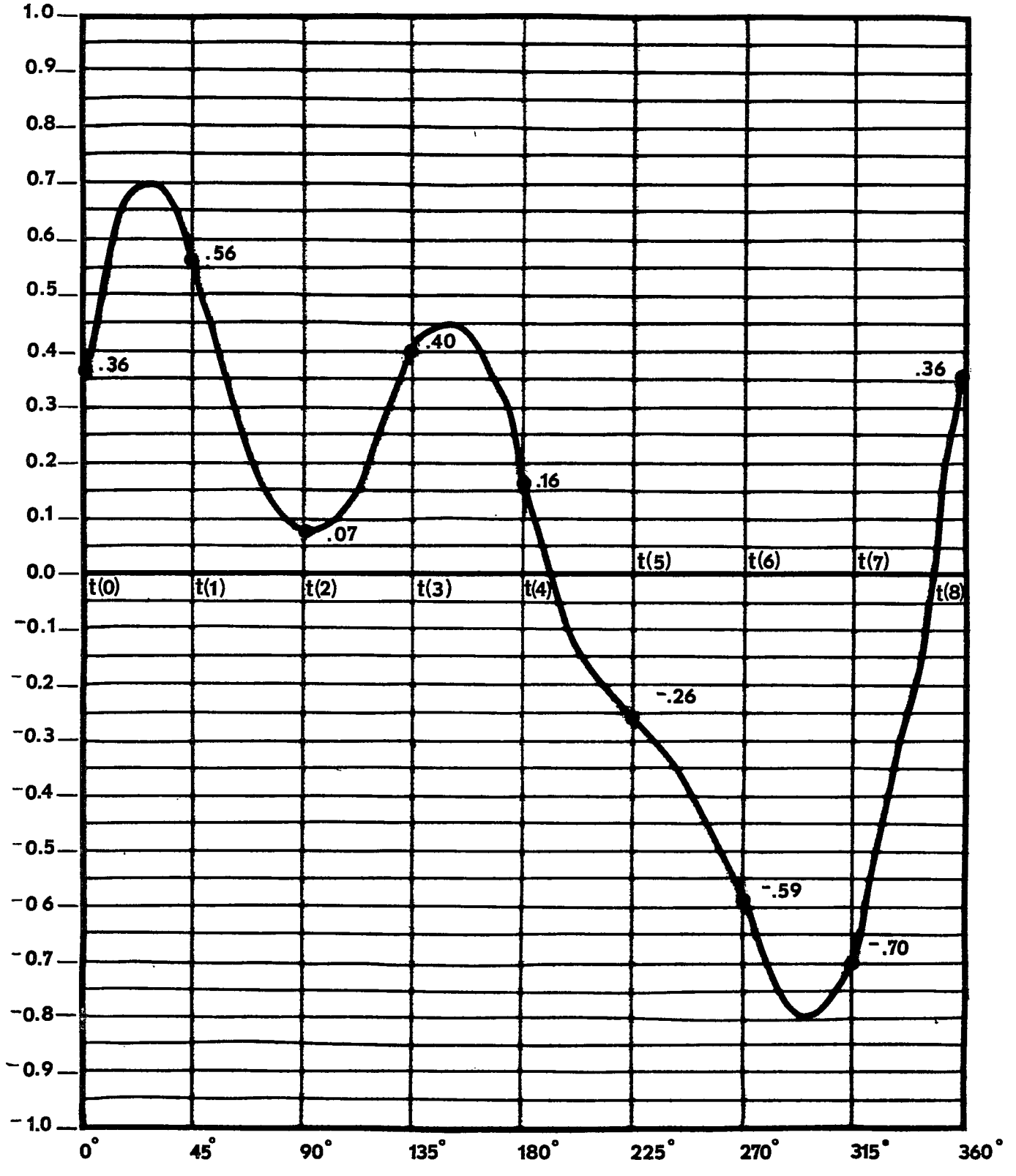
This is an informatory message which reports the number of equally spaced intervals into which a full cycle of the waveform must be divided. 'n' will be twice the number entered in response to 6.2.1 plus 2.

6.2.4) "t('n') = "

WAVE is requesting the instantaneous value of the waveform at the endpoint of interval 'n'. The instantaneous values may be entered using any convienient linear scale.

6.3) Example waveform analysis

The waveform on the following page will be analyzed in this example.


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
-0.9
-1.0
.36
.56
.07
.40
.16
-.26
-.59
-.70
.36
t(0)
t(1)
t(2)
t(3)
t(4)
t(5)
t(6)
t(7)
t(8)
0°
45°
90°
135°
180°
225°
270°
315°
360°

The waveform was generated from the equation:

$$f(x)=0.5*\mathrm{Sin}(1.0*x+0) + 0.3*\mathrm{Sin}(2.0*x+60) + 0.2*\mathrm{Sin}(3.0*x+30)$$

Note: x is in degrees, not radians.

Thus the harmonic components are:

Harmonic	Amplitude	Phase Shift
1	0.5	0.0
2	0.3	60.0
3	0.2	30.0

The waveform generation equation is given so that the results of the analysis can be compared to the generation equation. If the waveform generation equation were known, there would be no need for this analysis.

The number of harmonics considered and computed will both be 3. Thus the waveform will be broken into 8 equally spaced intervals 45 degrees apart. (One cycle is 360 degrees and 8 * 45 is 360.)

The instantaneous values of the waveform at the endpoints of the intervals are given below:

n	Computed t(n)	t(n) read from graph
0	f(0)=0.359808	0.36
1	f(45)=0.555318	0.56
2	f(90)=0.0669876	0.07
3	f(135)=0.396739	0.40
4	f(180)=0.159808	0.16
5	f(225)=-0.255317	-0.26
6	f(270)=-0.586604	-0.59
7	f(315)=-0.697639	-0.70
8	f(360)=0.359808	0.36

When the t(n)'s read from the graph are entered into the WAVE program, the following results are obtained:

Component	Harmonic	WAVE output	Actual value
Amplitude	1	0.513501	0.5
Phase Shift	1	0.625	0.0
Amplitude	2	0.30116	0.3
Phase Shift	2	59.5345	60.0
Amplitude	3	0.185339	0.2
Phase Shift	3	16.6768	30.0

The phase shift for the third harmonic appears to be a large error, but in fact it is less than a 4% error. WAVE's output is about 13 degrees lower than the actual value from the generation equation. This is an error of about 13 in 360, or less than a 4% error.

The more precisely the t(n)'s are known, the more precise the results of WAVE will be. Try entering the computed t(n)'s into WAVE and compare the results to the actual value.

7.0) Service

This section describes the procedure for reporting software problems and sending boards in for repair.

7.1) Reporting software problems

To report a software problem, the following information is necessary:

1) A complete description of the problem.

2) A floppy disk with a copy of the failing software and the source file(s) which cause the error.

3) Instructions on how to make the error occur.

4) The hardware configuration (size of memory, type of serial I/O, number of floppies, etc.) the music software is running under.

5) The release of the operating system being used.

6) The release of the music software being used.

Mail the above information to:

New Orleans General Data Services, Inc.
7230 Chadbourne Drive
New Orleans, Louisiana 70126

When the problem is resolved, your floppy will be returned with a copy of the latest distribution software (including a fix for the problem). (The original contents of the floppy will be over written.)

Package the floppy disk properly for mailing.

7.2) Repair and warranty

The HA8-2 board is fully warrantied against electrical failure for a period of 90 days after date of purchase. This warranty is limited to electrical failure of the HA8-2 board only, and does not cover physical damage to the music board or damages of any type whatsoever to devices connected to the HA8-2 including external amplifiers, the backplane or any other component of the computer system.

If the board needs repair, mail it postage paid to:

New Orleans General Data Services, Inc.
7230 Chadbourne Drive
New Orleans, Louisiana 70126

It is recommended that the board be packaged carefully and insured for mailing. If the board is under warranty, include a copy of the sales receipt showing date and place of purchase. Warranty work will not be performed unless a receipt is included.

The board will be repaired and returned via insured first class mail COD for parts, labor, and postage. Boards under warranty will not be charged for labor and parts, but will be charged for shipping and insurance. The current charge for labor is twenty five dollars per board. This charge is subject to change without notice. A request for a price quotation may be made before sending your board for repair.

The above repair policy applies only to electrical failures. Boards with physical damage will be repaired or returned without repair at the discretion of New Orleans General Data Services, Inc., depending on the severity of the physical damage.

8.0) The Song file

The song file is prepared using the text editor from sheet music. This section explains how to write the song file from sheet music. In the following description of the song file, lower case terms appearing in angle brackets are intended to be descriptive and are NOT to be included literally as part of the song file.

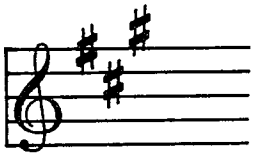
8.1) General format

A song file can be entered in free format. Spaces are treated as terminators, but are otherwise ignored. Multiple spaces can occur wherever a single space can occur. Horizontal tabs and line terminators (carriage return - line feeds) are treated in the same manner as spaces. There is no other significance to the end of a line. Other terminator characters (such as semicolons, parentheses, and brackets) can be preceded by spaces if desired - spaces are not required however.

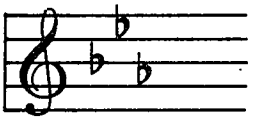
8.2) Key signature

The song file is composed of two parts, the key signature and the song text. The key signature indicates which notes are to be sharpened or flattened throughout the song. It consists of a sequence of sharps and flats separated by spaces and enclosed in parentheses. A sharp is a note letter ("A"-"G") immediately followed by the letter "S"; a flat is a note letter immediately followed by the letter "F".

For example, the key signature for A major,



would be written "(FS CS GS)". The key signature for Eb major,



would be written "(BF EF AF)".

Note that the key signature is required even if there are no sharps or flats in the key (e.g., C major or a minor). In these cases, the key signature is written "()".

8.3) Basic song text elements

The key signature is followed by the song text. The song text contains the notes that actually comprise the song. The text is made up of sequences of measures. Each measure contains an optional measure name followed by a measure body. The measure body is enclosed between "/" and "\":

<measure name> / <measure body> \

The last measure in the song is terminated by a double backslash:

<last measure name> / <last measure body> \\

The measure name, if included, consists of from 1 to 10 letters and/or digits.

The measure body is made up of from 1 to 4 voices. Every measure in the song must contain the same number of voices. Each measure need not have the same duration (duration is discussed below), but all voices in any given measure must have equal duration. Voices are separated by semicolons; e.g.:

/ <voice 1>; <voice 2>; <voice 3>; <voice 4> \

Voices consist of notes and rests. A note has three components: duration, note letter ("A"-"G"), and octave.

Duration indicates the length of time a note is played. Durations available are:

Song file Notation	Musical Notation
1	whole
2.	dotted half
2	half
4.	dotted quarter
4	quarter
8.	dotted eighth
8	eighth
16.	dotted sixteenth
16	sixteenth
32.	dotted thirty-second
32	thirty second
64.	dotted sixty-fourth
64	sixty fourth

If the duration is omitted when writing a note (or rest), the note is assumed to have the same duration of the previous note or rest in the same voice. This holds true even across measure boundaries. (The only exception is immediately after a "n-plet" as described below.)

Choosing a tempo that is too fast or too slow may make a duration too long or too short to be processed. A duration that is too long will result in an error message during song compilation. A duration that is too short will cause PLAY to stop when it is encountered.

A note letter is "A", "B", "C", "D", "E", "F", or "G" and indicates the pitch value of the note. A note letter can be optionally followed by an accidental mark, "F", "S", or "N", indicating flat, sharp, or natural. As in standard musical notation, an accidental applies to every succeeding instance of the note in the same octave for the remainder of the measure (unless offset by another accidental).

The octave of a note is indicated by a digit in the range 1 through 8. "1" represents the lowest octave available, with "A1" corresponding to the lowest "A" on the piano keyboard. Thus "C4" is middle C, etc. The seventh and eighth octave should be avoided because they are too high in frequency. From "G" in the seventh octave through the entire eighth octave, the pitch is too high to pass through the filters.

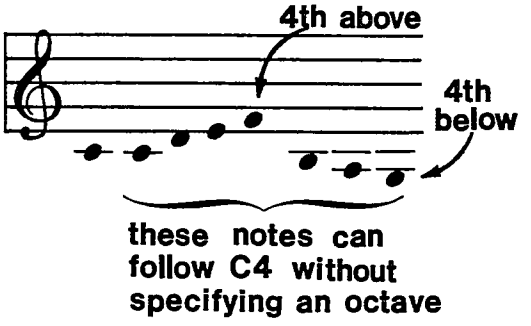
The octave of a note can be given in three ways: explicit, implied, or relative.

Explicit octave

The octave of a note can be explicitly given by following the note with one of the digits 1 through 8; e.g., "A1", "C4", "FS3" (third F sharp on the piano), "BF5" (fifth B flat), etc.

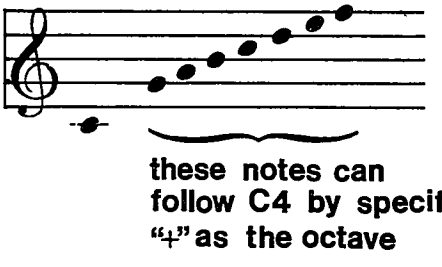
Implied octave

If the octave of a note is not explicitly specified, the note is assumed to be within a fourth above or a fourth below the preceding note in the same voice. This holds true across measure boundaries and across rests. For example, following a "C4" any of the notes "C4", "D4", "E4", "F4" (a fourth above), "B4", "A4", or "G3" (a fourth below) can be written without explicitly specifying the octave:

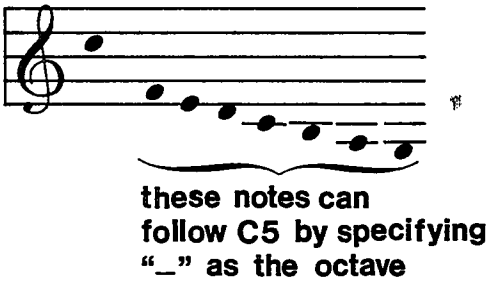


Relative octave

A note that is in the octave beginning a fifth above the preceding note of the voice can be indicated by following the note letter by a "+". Immediately following "C4" for instance, "G4" can be written as "G+"; "A5" can be written as "A+"; etc.



A note that is in the octave ending a fifth below the preceding note of the voice can be indicated by following the note letter with a "-":

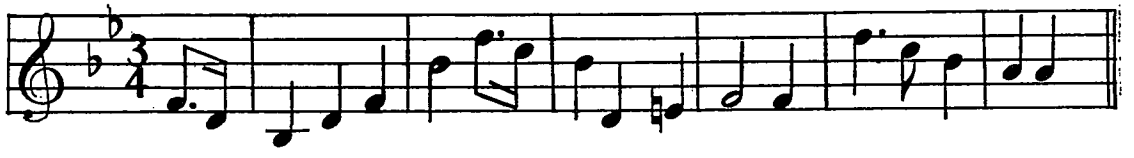


When writing a note, the duration, note letter, accidental mark, and octave are given in that order. No spaces can be

included.

Example

The following measures of the melody (one voice) can be notated as shown. (Line numbers are for reference only, and are not part of the song file.)



```
1.                (BF EF)
2.       MEAS1    /8.F4 16D\
3.                /4B D F\
4.                /2B 8.D 16C\
5.                /4B D- EN\
6.       MEAS5    /2F 4F\
7.                /4.D+ 8C 4B\
8.                /A A\\
```

Comments

1. Key signature contains B flat and E flat.

2. "MEAS1" is the measure name. The octave of the "D" need not be explicitly given since it is a third below the preceding "F".

3. The octave of these notes need not be given, since each is within a fourth of the previous. The duration ("4") of the "D" and "F" need not be given since it does not change.

4. Durations must be explicitly given for these notes.

5. Since the "D" is a sixth below the preceding "B", its octave is given as "-". "EN" means E natural - corresponding to the accidental.

6. This measure also has a name, "MEAS5".

7. The "D" is a sixth above the preceding "F"; thus, the octave is indicated by "+".

8. "\\" indicates the end of the song.

These measures could also be written explicitly giving

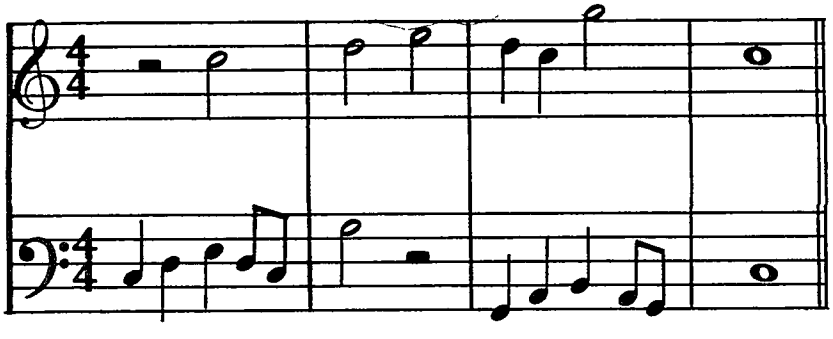
each duration and octave:

```
          (BF EF)
MEAS1     /8.F4 16D4\
          /4B4 4D4 4F4\
          /2B5 8.D5 16C5\
          /4B5 4D4 4EN4\
MEAS5     /2F4 4F4\
          /4.D5 8C5 4B5\
          /4A5 4A5\\
```

Rests are indicated by the letter "R" optionally preceded by a duration. As with notes, if the duration is omitted, it is assumed to be the same as the previous note or rest of the voice. Rests do not have accidental marks or octaves.

Example

The following example shows how two voices can be notated. (Line numbers are for reference only and are not part of the song file.)



```
1.        ()
2.        /2R C5;
3.         4C3 D E 8D C\
4.        /D E;
5.         2G+ R\
6.        /4D C 2G+;
7.         4G- A B 8A G\
8.        /1C-;
9.         1C\\
```

Comments:

1. Key signature indicates no sharps or flats.

2. First measure for voice. Since "C5" is a half note and the

preceding rest is a half rest, the duration ("2") can be omitted when writing the note.

3. First measure for second voice.

4. The full notation for the "D" would be "2D5". The duration ("2") can be omitted since the previous note in this voice ("C5") is a half note. The octave ("5") can be omitted since the "D" is within a fourth of the preceding note of the voice ("C5").

5. The duration of the "G" must be given, since the preceding note was an eighth note. Also, since this "G" is a fifth above the preceding "C", the octave is specified as "+". The note could have also been written as "2G3".

6. The "2G+" here could have been written as "2G5".

7. The octave for the "G" here is given as "-". The preceding note is the half note "G" at the start of the preceding measure. This note is an octave above this "G". The duration ("4") must be given since the rest preceding it is a half rest.

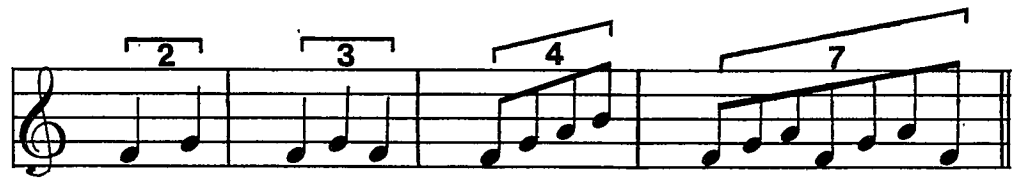
8. Whole note "C5".

9. Whole note "C3".

8.4) Duplets, Triplets, etc.

A duplet is two notes played for a total time duration equivalent to three. A triplet is three notes played in the time normally taken by two. A quadruplet is four notes played in the time of three. For 4<n<=7, an n-plet is n notes played in the time of four. (The user should be aware that this convention may not coincide with musical usage in all instances.)

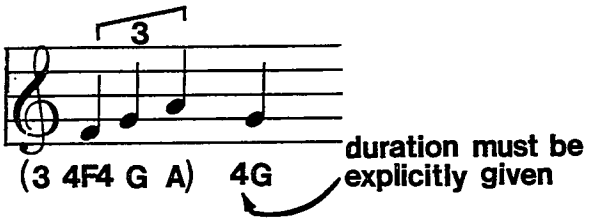
A duplet must contain exactly two notes of equal duration; a triplet must contain exactly three notes; in general an n-plet (2<=n<=7) must contain exactly n notes of equal duration. (However these notes can be tied as explained below.) N-plets are written in parentheses, with the number of notes in the n-plet as the first item:



(2 4F4 G) (3 4F G F) (4 8F G A B) (7 8F G A F G A F)

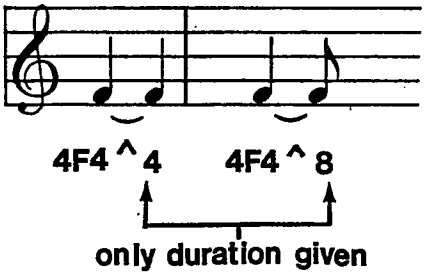
N-plet	Song file notation
Duplet	(2 4F4 G)
Triplet	(3 4F G F)
Quadruplet	(4 8F G A B)
Septuplet	(7 8F G A F G A F)

"Duration of the preceding note" is not defined following a n-plet. Therefore, an explicit duration should always be given for the note following a n-plet.

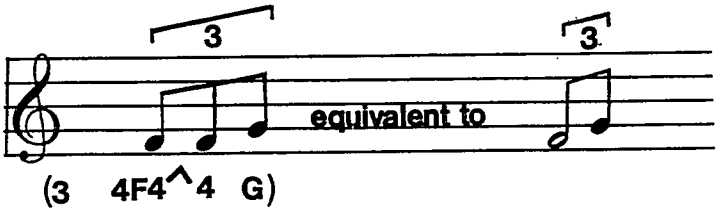


8.5) Tied notes

Notes of identical pitch can be tied. Tying notes causes the notes to be sounded without a separating pause. Thus when two quarter notes of the same pitch are tied, they are sounded as a half note. When notes are tied together, the pitch (note letter, accidental, octave) is given only for the first note. Succeeding notes are indicated by giving their duration only. Tied notes are connected by carets (^) (or up-arrows):

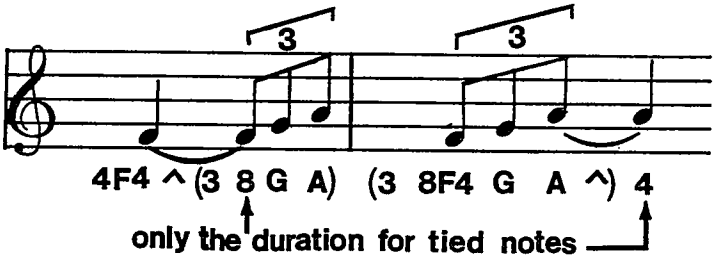


Notes can be tied within n-plets (Recall in song file notation, a triplet must contain exactly three notes even though these can be tied):

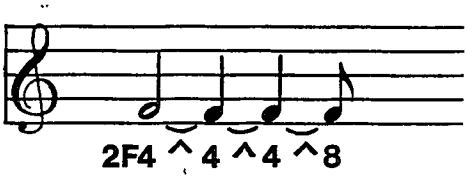


"(3 2F4 G)" is not legal since the triplet here contains only two notes. This must be written "(3 4F4^4 G)" as shown above.

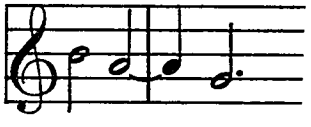
Notes can be tied across n-plets by writing the caret before the open or close parenthesis:



More than two notes can also be tied:



The only exception to the rule of writing only the duration of a tied note is the case in which a note is tied across a measure. In this instance, the caret immediately precedes the semicolon or backslash that ends the measure for that voice. The first note of the voice in the next measure is notated exactly as if it were not tied:



is written:
```
/2B5 A^\
/4A 2.G\
```

and not:
```
/2B5 A^\
/4 2.G\
```

8.6) Repeats

Several methods are available for indicating repetition in the song file. A colon before the final backslash of a measure (":\") indicates repetition. The repetition can begin either:

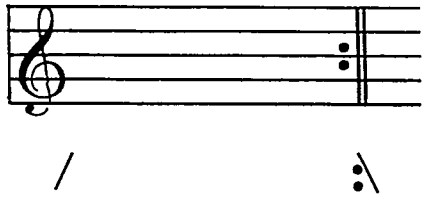
1. At the matching preceding colon coming after a measure opening slash ("/:"); or

2. At the beginning of the song if there is no matching "/:".

In the following example all of the music (any number of measures) between "/:" and ":\" is to be repeated:.

/: :\

In the following example, everything from the beginning to the ":\" is to be repeated:



/ :\

The following examples illustrate the order that repeated measures are played:

Example 1:

/ <a> \ /: <b> :\ / <c> \\

will be played:

<a> <b> <b> <c>

Example 2:

/ <a> :\ / <b> \\

will be played:

<a> <a> <b>

Example 3:

/: <a> :\ /: <b> :\\

will be played:

<a> <a> <b> <b>

Repeats are played only the first time that they are encountered:

Example 4:

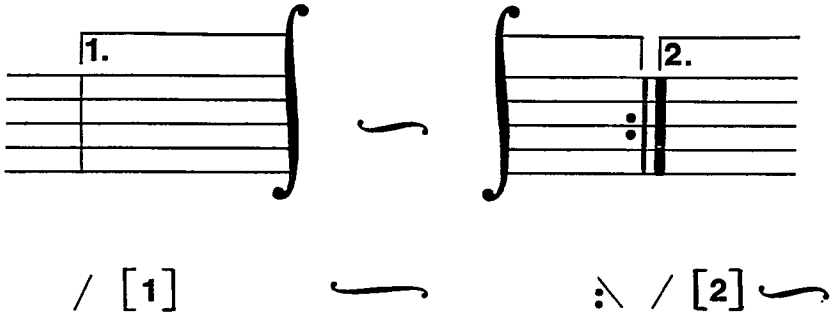
/: <a> :\ / <b> :\\

will be played:

<a> <a> <b> <a> <b>

8.7) First and second endings

First and second endings are indicated by "[1]" and "[2]" respectively at the beginning of a measure:



A repeat must occur between a first and second ending. It need not be in the same measure as the first ending. The first ending is played the first time through. When the repeat is taken, the first ending is omitted and play continues with the second ending.

Example:

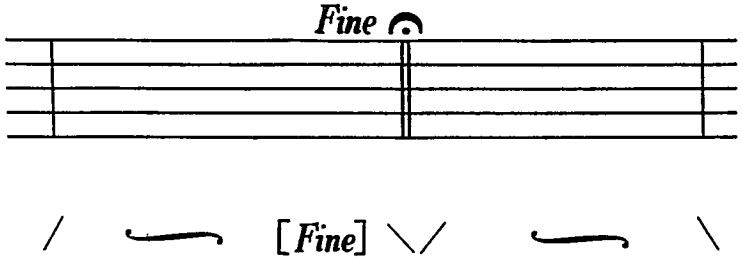
/ <a> \ /[1] <b> :\ /[2] <c> \\

will be played:

<a> <b> <a> <c>

8.8) Fine

"[FINE]" can be placed at the end of a measure. It is ignored the first time it is reached. The song will end the second time "[FINE]" is encountered.



Example:

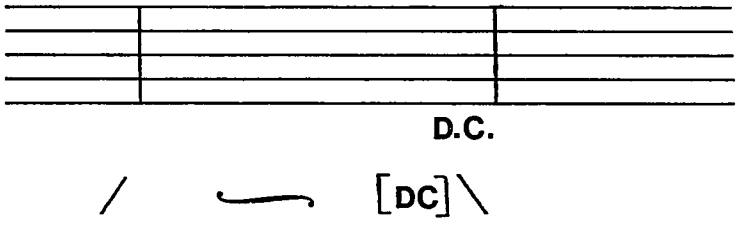
/ <a> [FINE]\ / <b> :\\

will be played:

<a> <b> <a>

8.9 Da capo and dal segno

The designation "[DC]" can be written at the end of a measure. It will cause the song to be repeated from the beginning. When playing a repetition caused by "[DC]", other repeats are ignored. The song will end when "[FINE]" or "\\" (end of song text) is encountered.



Example 1:

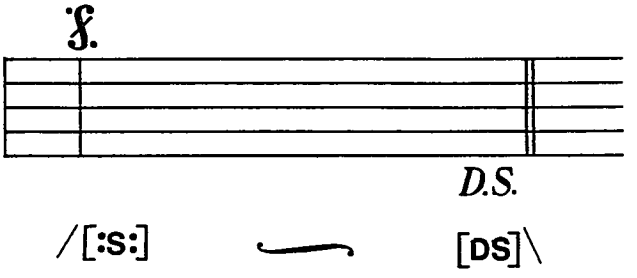
/ <a> [FINE]\ / <b> [DC]\\

will be played:

<a> <b> <a>

The designation "[DS]" can also be written at the end of a measure. "[DS]" is similar to "[DC]", except that repetition

starts at the nearest preceding measure beginning with "[:S:]" (segno), rather than at the beginning of the song.



Example 2:

/ <a> \ /[:S:] <b> \ / <c> [DS]\\

will be played:

<a> <b> <c> <b> <c>

Example 3:

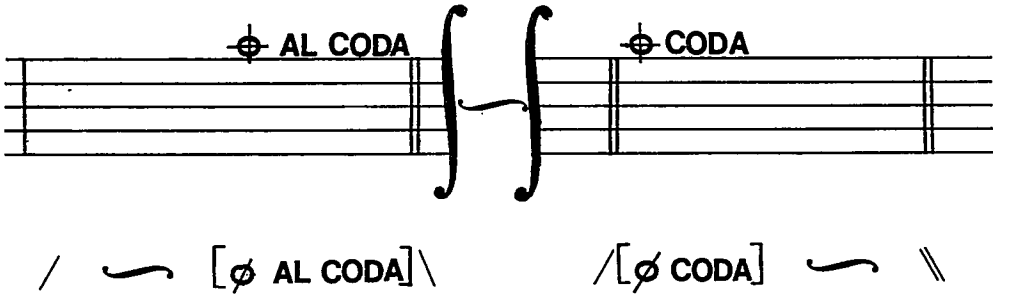
/ <a> \ /[:S:] <b> \ / <c> [FINE]\ / <d> [DS]\\

will be played:

<a> <b> <c> <d> <b> <c>

8.10) Coda

The designation "[O AL CODA]" can appear at the end of a measure. It is ignored the first time it is encountered. When encountered upon a repetition, it causes play to continue with the nearest following measure beginning with [O CODA]".



Example:

/ <a> [O AL CODA]\ / <b> :\ /[O CODA] <c> \\

will be played:

<a> <b> <a> <c>

If a measure contains more than one of the symbols, the order in which they appear is irrelevant; e.g. "/:[:S:]" is equivalent to "/[:S:]:".

8.11) Miscellaneous

Dynamics

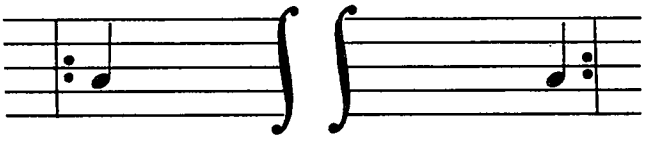
There are no dynamics provided by the system.

Staccato

All notes are played legato - with no break between notes. The only exception is when the same note is repeated. In this case, a brief pause will separate the notes so that they are clearly articulated as individual notes. A staccato-like effect can be achieved by replacing a note by a shorter note plus a rest.

Same pitch in a repeated section

If a repeated section ends with the same note as it begins, e.g.



a gap will not be automatically inserted to separate the notes. It may be desireable to shorten one of the notes and add a rest to achieve proper articulation. For example,

Inserting comments in the song file text

The compiler ignores any song file text enclosed in single quotes. This can be used to add comments or in debugging the song. If a portion of the song has to be debugged, unwanted measures can be enclosed in single quotes. The quoted measures will be taken as comments by the compiler and just the portion to be debugged will be compiled. The duration, pitch, and octave must be specified explicitly for the first measure after a comment. Since the context is usually derived from the previous measure it will not be as intended (The previous measure has been made a comment). The commented measures may not contain comments because the first quote forming the inner comment will cause a termination of the quote and subsequent text will not be interpreted as a comment.

8.12) Error messages

COMPOSE error messages for the song file will provide the number of the measure in which the error occurred and the name of the measure, if present.

Most error messages are self explanatory. The following list indicates remedies where possible.

8.12.1) "?Duration expected in n-plet"

Only a duration is allowed following a tie, even when the tie is across an n-plet.

e.g. "4B^(3 8B C D)" is illegal;
"4B^(3 8 C D)" is correct.

See section 8.5 "Tied notes" for more information.

8.12.2 "?Duration of n-plet too long"

Total length of a n-plet is too long to be processed. Try breaking it into shorter n-plets if possible.

e.g. "(6 1A B C D E F)" can be written
"(3 1A B C) (3 1D E F)".

8.12.3) "?Illegal character"

An invalid character has been scanned. This could be an illegal note, octave, duration, etc.

8.12.4) "?Illegal duration"

Duration of a note is not a legal value. See section 8.3 "Basic song text elements" for more information.

8.12.5) "?Illegal key signature"

Key signature is missing or invalid. Recall that if there are no sharps or flats, the key signature must still be given as "()". See section 8.2 "Key signature".

8.12.6) "?Illegal prefix at start of measure"

Entries in brackets at the beginning of a measure can be only "[0 CODA]", "[1]", "[2]", or "[:S:]".

8.12.7) "?Illegal symbol at end of measure"

Entries in brackets at the end of a measure can be only "[FINE]", "[DS]", "[DC]", or "[0 AL CODA]".

8.12.8) "?Illegal token in a voice"

An unexpected "/" or ")" has been encountered; or a tie ("^") is not followed by a legal token (i.e., a duration, semicolon, backslash, open parenthesis, open bracket, or colon).

8.12.9) "?Illegal symbol at end of measure"

Measure must end with a "\". Only a colon (repeat), "[FINE]", "[DC]", "[DS]", or "[0 AL CODA]" can occur between the last note of the measure and "\".

8.12.10) "?Measure too long"

Total duration for a measure is too long. Break the measure into shorter measures.

8.12.11) "?More than one repeat at end of measure"

Only one repeat symbol (colon) can appear between "/" and the first note of a measure.

8.12.12) "?More than one repeat at start of measure"

Only one repeat symbol (colon) can appear between the last note of a measure and the "\".

8.12.13) "?No close paren or too many notes in n-plet"

A n-plet must contain exactly as many notes as indicated by the number at the beginning of the n-plet. A duplet must contain two notes; a triplet must contain three notes, etc. See section 8.4 "Duplets, Triplets, etc" for more information.

8.12.14) "?No coda found after [0 AL CODA]"

If the end of a measure contains "[0 AL CODA]", then a later measure must contain "[0 CODA]" at its start. See section 8.10 "Coda" for more information.

8.12.15) "?No number found in n-plet"

A n-plet must contain a number between 2 and 7 (inclusive) immediately after the open parenthesis. See section 8.4 "Duplets, Triplets, etc".

8.12.16) "?No second ending found"

If a measure contains "[1]" (first ending) at its start, a later measure must contain "[2]" (second ending). See section 8.7 "First and second endings" for more information.

8.12.17) "?No signa found after DS"

If the end of a measure contains "[DS]", then a previous measure must contain "[:S:]" (signa) at its start. See section 8.9 "Da capo and dal segno" for more information.

8.12.18) "?No slash beginning measure"

Measure must start with a "/".

8.12.19) "?Note expected at start of n-plet"

A note token must follow the number at the start of a n-plet unless a tie occurs immediately before the n-plet. (the n-plet is "tied into"). See section 8.4 "Duplets, Triplets, etc" and section 8.5 "Tied notes" for more information.

8.12.20) "?Note too long"

The duration of a note is too long. Other than playing the song at faster tempo, the only way to fix this is to end the measure in the middle of the note, break the note into two parts, one ending a measure, the other starting the next measure, and tie the two parts together across the (new) measure boundary.

8.12.21) "?N-plet number too large"

The largest n-plet that can be handled is a septuplet. See section 8.4 "Duplets, Triplets, etc" for more information.

8.12.22) "?Second ending without proper first"

If a measure contains "[2]" (second ending) at its start, then a preceding measure must contain "[1]" (first ending), and a repeat (":\") must occur between the "[1]" and the "[2]". See section 8.7 "First and second endings" for more information.

8.12.23) "?Song too long"

Song is too long to be handled by system. Can it be broken down into shorter movements?

8.12.24) "?This message should never appear"

It shouldn't. However, if it does it indicates an internal error. Please report the problem. For more information, refer to section 7.1 "Reporting software problems".

8.12.25) "?Too few voices in measure"

Every measure must contain the same number of voices as indicated during the initialization phase of COMPOSE. "Silent" voices can be filled in with rests if necessary.

8.12.26) "?Too many accidentals in measure"

An internal system limit has been exceeded. Break the measure into two smaller measures.

8.12.27) "?Too many errors"

Too many errors have been encountered by COMPOSE and it gives up trying to compile the song file. Correct the errors detected before this message and try again.

8.12.28) "?Too many notes in measure"

An internal system limit has been exceeded. Break the measure into two smaller measures.

8.12.29) "?Too many repeats, DC etc"

An internal system limit has been exceeded. Try explicitly writing out some of the repeated measures. The editor can be used to do this relatively easily.

8.12.30) "?Too many voices in measure"

Every measure must contain the same number of voices as indicated during the initialization of COMPOSE.

8.12.31) "?Two first endings"

Only one measure starting with "[1]" (first ending) is permitted.

8.12.32) "?Voices not all of equal length"

Every voice in a given measure must have the same total duration. Fill in "silent" voices with rests if necessary.

Appendix A

Pitch	Frequency	Maximum Harmonic		Pitch	Frequency	Maximum Harmonic
A1	27.5	9		A5	440	6
AS1	29.1352	9		AS5	466.164	6
B1	30.8677	9		B5	493.883	6
C1	32.7032	9	HC	C5	523.251	5
CS1	34.6478	9		CS5	554.365	5
D1	36.7081	9		D5	587.329	5
DS1	38.8908	9		DS5	622.253	4
E1	41.2034	9		E5	659.254	4
F1	43.6535	9		F5	698.456	4
FS1	46.2493	9		FS5	739.988	4
G1	48.9993	9		G5	783.99	3
GS1	51.9129	9		GS5	830.607	3
A2	55	9		A6	880	3
AS2	58.2705	9		AS6	932.328	3
B2	61.7354	9		B6	987.767	3
C2	65.4064	9		C6	1046.5	2
CS2	69.2957	9		CS6	1108.73	2
D2	73.4161	9		D6	1174.66	2
DS2	77.7816	9		DS6	1244.51	2
E2	82.4068	9		E6	1318.51	2
F2	87.307	9		F6	1396.91	2
FS2	92.4985	9		FS6	1479.98	2
G2	97.9987	9		G6	1567.98	1
GS2	103.826	9		GS6	1661.21	1
A3	110	9		A7	1760	1
AS3	116.541	9		AS7	1864.66	1
B3	123.471	9		B7	1975.53	1
C3	130.813	9		C7	2093	1
CS3	138.591	9		CS7	2217.46	1
D3	146.832	9		D7	2349.32	1
DS3	155.563	9		DS7	2489.01	1
E3	164.814	9		E7	2637.02	1
F3	174.614	9		F7	2793.82	1
FS3	184.997	9		FS7	2959.95	1
G3	195.997	9		G7	3135.96	0
GS3	207.652	9		GS7	3322.43	0
A4	220	9		A8	3520	0
AS4	233.082	9		AS8	3729.31	0
B4	246.942	9		B8	3951.07	0
MIDDLE C C4	261.626	9		C8	4186.01	0
CS4	277.183	9		CS8	4434.92	0
D4	293.665	9		D8	4698.63	0
DS4	311.127	9		DS8	4978.03	0
E4	329.627	9		E8	5274.04	0
F4	349.228	8		F8	5587.65	0
FS4	369.994	8		FS8	5919.9	0
G4	391.995	7		G8	6271.92	0
GS4	415.304	7		GS8	6644.86	0