**ASLP EDITOR AND ASLP COPYRIGHT 2012 DSAPSC. ALL RIGHTS RESERVED**

**The ASLP editor is provided as freeware. It is not to be sold nor distributed without prior permission from DSAPSC. It is free to use in non commercial projects and for home enjoyment. If a user wishes to use the player in a commercial project then please contact me at DSAPSC.COM so we can work out an agreement.**

**WHAT IS THE ASLP ??**

The **A**dvanced **S**ound **L**ist **P**layer is a program that reads a user supplied byte code and generates sounds from it. It recognizes both musical notes and various commands designed to assist in creating music or sound effects. The source code is available for inclusion in your assembly language programs to assist in producing high quality sounds. The player reads the lists produced by the Editor and plays them in real time via a user supplied timing mechanism (either the VDP interrupt or the 9901 timer.)

**WHAT IS THE ASLP EDITOR ??**

The ASLP Editor is a program designed to assist the user in creating the lists used by the ASLP. It is analogous to the assembly language line by line assembler and uses mnemonics to represent commands and notes. It is menu driven with reasonably easy to understand commands and note syntax so memorization of commands is not needed. Included within the editor is the ASLP itself so that you may listen to your creation as you build it. The editor assembles your commands on the fly. There is no lengthy second step.

It is important to understand that the editor and player are completely separate entities. When a selection is played the editor simply points to the list(s) and passes control to the player. **WHAT EVER IS CURRENTLY IN THE LIST IS PROCESSED BY THE PLAYER.** Commands that were previously in the list but removed at some point simply do not exist anymore. This causes some confusion to the new user as the tendency is to look at the editor as a control panel for the player. **WHEN EVER THE PLAYER IS INVOKED, ALL THE DEFAULTS ARE SET TO A NEUTRAL STATE. ALL OF THE SOUND PARAMETERS NEED TO BE SET UP WITHIN THE LIST.**

Some common mistakes incurred by first time users are….

1. Neglecting to set a volume for the list. When the player is invoked the volume of the voice is turned completely off. It will need to be set with the VOL command before sound can be heard.
2. Neglecting to set a tempo. The TEM command builds the duration table when it is encountered. If the TEM command is not executed then the player does not know how long a duration should be and it will not perform correctly.
3. Deleting a VOL or TEM command from the list. Remember that the player gets its information only from the current list in memory.
4. Setting a command in 1 list and expecting it to affect the other lists. A VOL command in list 1 **only** affects the volume of list 1. Each list needs its own set of commands**. The only exceptions to this rule are the TEM command and the CAL command.**
5. Executing a special effect command such as vibrato and not turning it off when the effect is no longer desired.

A short list to play a simple scale would be…

**VOL-0 TEM-2 C2 D2 E2 F2 G2 A2 B2 END (inserted by the editor.)**

The player will insert the end command for you and when it is encountered the volume is turned off. When the ASLP is invoked from the editor then the defaults are also reset. This prevents commands that affected the end of the sound list from appearing at the beginning of the next playing and creating confusion.

The following section deals with descriptions of the various functions of the ASLP Editor.

**MAIN SCREEN**

When the main screen is up you can do one of three things. JOY-up/down will change the current tone step which is displayed near the middle of the screen. Pressing the fire button will begin the note entry process. Pressing a hot key will take you to a current menu. Utilizing the left/right function of the joystick will scroll through the list at the bottom of the screen.

Available tones are those from A0 (a octave zero) to A7. This notation was selected instead of a normal scale to maintain consistency with TI’s odd choice. The editor will not let you scroll above or below these values.

Pressing the fire button will begin the note entry process.

**NOTE ENTRY**

During note entry a sub menu will pop up which contains several sub entries the first of which is DURATION. The duration describes the note type and can either be Whole, Half, Quarter, Eighth, Sixteenth or Utility durations. They are changed by the joystick left or right and selected by the fire button. NOTE: Not all tempos support sixteenth notes and the editor will not allow them to be entered in those durations. The utility duration will be discussed later.

Next are the modifiers. A note may be dotted by selecting the appropriate Y/N answer which is controlled by the joystick left/right function. A note may be tied to the next by the same process. Finally your note may be modified further in the last step. The MODIFIER entry can make a note a rest, a sharp or do nothing. Using the joystick the options of “ “,”R” and “#” appear. Press the fire button to select one. It needs to be noted that flats are not offered as they can be described as sharps and not all notes have sharps. The editor will not offer the sharp option on these notes. The editor does not support flatted notes. In order to create a flatted note you must go down to the next full note and make it sharp. It produces the same tone.

Once the last modifier is selected the note will be inserted into the list. At any point you may press FCTN-4 to abort the entire entry

**Special note for voice 4**: Voice #4 can only play noise values. Although the notes are entered as usual the entry will be automatically specified as a noise entry and the sharp modifier will be ignored. Dots, Ties and rests are still valid.

**HOT KEYS**

**INSERT FUNCTION (I)**

If you find you need to insert a command into the list then you may use the joystick to scroll to the command that will be right after the one you wish to insert. At this point press ‘I’ and the list shall be shifted to the right 1 position and the NOP command will be inserted. NOP is exactly what it sounds like. It is a valid command that really only holds a place in the list. It may be changed into a note or command at any time or left in. The player will recognize it, more or less skipping it in real time.

**DELETE FUNCTION (D)**

Using the joy left/right function, scroll to the command you wish to delete and press ‘D’. The command will be deleted and the list shifted to the left. There is no undo command so be careful.

**CHANGE VOICE (V)**

The editor is capable of editing and building four list, one for each voice. Only one list may be edited at a time though. Pressing ‘V’ will prompt a sub menu to appear asking you which voice (actually voice list) that you wish to work on. At this time press the numbers 1,2,3 or 4 followed by ‘Enter’. The current list being worked on is displayed on lower right, below the main window.

**PLAY LIST (P)**

The editor has the ability to play each voice independently or all 4 voices together. The submenu will ask you to press 1,2,3,4 or A. Once play has begun pressing FCTN-4 will abort playing. When the list is finished the program will return to the main window.

**FILE (F)**

There are three sub functions in the files section.

**(L)OAD** loads an ASLP file into memory from a DSK device.

**(S)AVE** saves an ASLP file to a DSK device.

ASLP files are always 8K in length. This doesn’t mean your lists are 8K but rather the entire list space (8K) is saved to disk.

**( C)ONVERT**- converts an ASLP file into a DF-80 file composed of labels and DATA statements. When a file is converted all the measure markers and NOPs are removed. Only the data that compose the lists are included in the source (with one minor exception.) Even if you are only working with 1 list, all four of them are converted. Please see note (A) at the end of this document for a full description of converting ASLP files to assembler source code.

**MARKER (M)**

Pressing M will insert a numbered marker into the list. The marker has not effect on playback but is only there as a visual cue to make tracking easier. Markers are always numbered from 1 forward. If you delete or insert a marker then they will all be renumbered starting with the first one being designated as number 1. Each list has its own set of markers.

**COMMAND ( C )**

Pressing ‘C’ will take you to the command window. Commands are used to tell the player to do specific things during list processing. They are executed in real time. Once the command window is entered, navigation is made with the joystick. An arrow will appear in front of the current command and pressing fire will enter its sub window. Changing values in commands is always achieved through the Joystick left/right. When satisfied with the value then press the fire button. The command will be entered in the list and the sub menu will close. To abort either a sub window or the command window press FCTN-4. Below are a list of the legal commands and what they do.

**TEMPO (TEM)**

The Tempo command sets the stage for the player and must always appear before a note is played. It sets the durations for all the notes. Without it the player assumes that all whole notes are 255 VDP ticks in duration. Changing the tempo value will alter the display in several ways. It shows you how many VDP cycles a whole note is and what notes are legal. IE Tempo 0 has a whole note value of 8 with legal notes of W, H, Q and E. Sixteenth notes are not legal in this tempo and the editor will not allow them.

Tempos are defined based on the value of the whole note NOT a traditional beats per minute fashion as in musical notation. If a tempo gives a whole note a value of 64 then that means a whole note will last for 1.06 seconds . The value of the half note will be 32 (64/2) etc. If during this division an eight note is odd, then the sixteenth notes will not be an integer and therefore do not fit into the scheme.

It is best to make TEMPO you first command in list one. TEMPO is a global setting determining the settings for all four lists. It can be changed at any time but be warned. **If you have more than one list playing and they are not playing identical notes then they may become out of sync with each other. The TEMPO command is the only command that affects all lists.**

**VOLUME (VOL)**

The Volume command determines the volume for a particular list. It needs to be specified before notes are played as the editor assumes it is off. The specification (ranges from 0 volume full on to 15 completely off.) **Remember the volume needs to be set for every list**.

**VOLUME BUMP (BMP)**

Volume bump simple bumps the volume of the list up or down one notch. The player keeps track of the current volume and does not let it wrap around.

**RELEASE (REL)**

The release command adds a little bit of volume envelope control to the sound. When a release is specified the tone will begin to fade away towards the end instead of blending into the next one. For example if you are playing a whole note with a value of 32, a volume of 0 and a release value of 16 then the note will play full volume for 16 VDP ticks and then decrease in volume by one every VDP tick after that. The effect is that two identical notes played back to back will be heard as two notes instead of one long tone.

**DETUNE (DET)**

The Detune command allows you to fudge the absolute frequencies of the player. Legal values are -128 to 127. This value is added to the tone in real time to change its frequency. It should be understood that because of the nature of the 9919 chip (a design only an EE could love)the negative values actually increase the frequency while the positive ones decrease it. To turn Detune off specify a value of 0.

**UTILITY DURATION (UTL)**

The utility duration is a special duration that is user defined and may be changed throughout the list. Its legal values are 1 through 255. A note with a duration of ‘U’ will play for the length specified by this command. Tempos such as #7 (whole note value of 64 and an even numbered sixteenth note) can have 32nd notes added by specifying a utility duration of ½ a sixteenth note.

**NOISE (NOI)**

The noise command specifies the noise that is currently available for voice 4. It may be changed at any time. It should be noted that noises can still utilized some of the player commands such as release and volume bump. Tone specific commands (although allowed) will have no effect.

**VIBRATO (VIB)**

The Vibrato command allows you to create a vibrato effect during the playing of the tone. The first value of vibrato is the amplitude of the effect. This number will be added to the tone every other instance. The second number is the frequency that the vibrato occurs in VDP ticks. If the amplitude is specified as 1 and the frequency is two then the player will alternate between true frequency and frequency plus 1 every other VDP tick. To turn vibrato off specify a zero for both values

**FORCE TONE (FTN)**

The Force Tone command is used in conjunction with the Portamento command. It is used to make the player thank that the last note it played is something other than what it actually played. The two entries are the actual musical half step you desire and the octave.

**PORTAMENTO (POR)**

The Portamento command creates a glide between notes being played. A good example is a slide whistle. Instead of jumping from C1 to C2 the player with increase the frequency each VDP tick until C2 is reached. It works automatically up or down so the entry is always a positive value. A value of 0 turns Portemento off. **It is important to understand that the starting point for portamento processing is always the last note played. This does not always produce the desired results therefore the force tone command is used to trick the player into changing the “last note played.”**

**DEFINE PHRASE (DEF)**

A phrase is a group of frequently played sounds that can be accessed from anywhere in the list or in other lists as well. It is analogous to a subroutine in Extended Basic. There are 16 available phrases that can be used. That is to say sixteen total phrases and not sixteen per list. Each phrase must have a unique number and once defined may not be redefined. The editor will not allow you to redefine a phrase. Although you may define a phrase inside another phrase the results are usually not good.

**RETURN FROM PHRASE (RET)**

The RET command does just what it sounds like. It returns to the command directly after the one that called the phrase UNLESS the phrase was not called but rather processed for the first time. At that point is ignored.

**CALL PHRASE (CAL)**

CAL simply calls the phrase which is then played. Calls made be made from anywhere in the list (or other lists) except from within phrases. List 3 may use a phrase defined on list 1 provided it has all ready been defined.

**IMPORTANT RULES FOR PHRASING……**

1. **Do not define a phrase within another phrase. IE DEF…..DEF….RET RET will kill the player**
2. **Do not call a phrase that has not been defined**
3. **After you define a phrase you must end it with a RET command before you define another**

**If the player locks up when you are using phrases it is likely one of these rules has been broken. I plan to add some error checking to the editor later to hopefully prevent this annoying feature. Please be careful using phrases.**

**BEGIN REPEAT HEAD (HED)**

This command specifies the number of times the next section of list shall be repeated. It is used in conjunction with the TAL command.

**END REPEAT TAIL (TAL)**

This command ends the repeated section. In a repeat section an example sequence would be…

1. **HED 16 repeat the section 16 times**
2. **Various commands**
3. **TAL decrement the counter and repeat the commands if needed. Otherwise drop through**

**A repeated section is like a FOR/NEXT loop.**

**Common reasons the player does not produce sound….**

1. **No volume specified (editor always assumes the volume is off at invocation.)**
2. **No Tempo Specified (the TEM command actually builds the note table.)**

**Common reasons that portamento does not work right…**

1. **The last tone played is inconsistent with the effect you desire. See Force Tone.**

**TEMPOS SUPPORTED BY THE ASLP AND THEIR NOTE DURATIONS…**

**TEMPO W H Q E S U32nd**

**0 8 4 2 1 - -**

**1 16 8 4 2 1 -**

**2 24 12 6 3 - -**

**3 32 16 8 4 2 1**

**4 40 20 10 5 - -**

**5 48 24 12 6 3 -**

**6 56 28 14 7 - -**

**7 64 32 16 8 4 2**

8 72 36 18 9 - -

9 80 40 20 10 50 -

10 88 44 22 11 - -

11 96 48 24 12 6 3

12 104 52 26 13 - -

13 112 56 28 14 7 -

14 120 60 30 15 - -

15 128 64 32 16 8 4

**Note about tempos.** Other tempos may be introduced into the player by changing the ASLP source code. The Table is labeled TTABLE and contains the whole note values for the specified tempo. This may be changed by the user to something different provided that all the rules governing tempo are followed. Legal values are 1 to 255.

**NOTE ON CONVERTING ASLP FILES TO ASSEMBLY SOURCE.**

When you convert an ASLP file to an assembly source file the editor creates a DF80 file containing only the data within the lists so it may be drastically shorter than the 8K of the list. The format of the DF80 file is always a label followed by data statements composing the list. This is repeated four times, once for each list. The labels are always “LIST1”, LIST2”, “LIST3” and “LIST4”.

At this time the editor does not remove NOPs or Measure Markers from the source code. Additionally the last line of a list may contain some garbage. This will be assembled but does not affect the player as it always occurs after the list END statement. If space is at a premium then they may be removed. The garbage always appears after and on the same line as a >0700.

Currently the conversion process creates a DF80 type file which is not usable by the assembler in the EA module. To make is usable simply load the converted file into the EA editor and resave it. This will create a DV80 file which may be assembled.

After loading the df80 file into the EA editor it is possible to change the label names to something meaningful, delete unused lists and remove any garbage statements. When done simply save the file and it is ready to be included into your assembly source code along with the player.

As of this 11/11/12 I plan on modifying the editor to automatically remove the garbage, NOPs and Measure Marker statements prior to saving the lists as source code. Additionally I may prompt for label names and desired lists to save. We’ll see.