

Chapter 16 Note Sequencer

The note sequencer has been modified so that it is easier for you to create and use sequences in your programs. To do this we have added support for sequences that are not located at fixed memory locations. So, you can now create a handle to build or load your sequence into. We have also now added a new sequence command that can call any user routine from inside a playing sequence.

New Calls

StartSeqRel	Call Number	\$15
input	ErrHndlrRoutine	LONG
	CompRoutine	LONG
	Sequence	HANDLE
output	none	

This call is the same as the StartSeq call, with the following exceptions:

1. The Address that the Handle is pointing to is used as a the BaseAddress.
2. All pointers to Phrases and Patterns are now treated as relative pointers. This means that the pointers in a phrase are now added to the BaseAddress supplied, and that is used as the address for the new data structure.

Up until now the Note Sequencer only supported absolute addresses in a phrase. This ment that you could not design a relocatable sequence.

Now you can use relative addresses instead of absolute addresses in a phrase, if you make the StartSeqRel call instead of the StartSeq call. The note sequencer will then add your relative address to the base address and use that as the address to your next phrase or pattern. There is no checking for overflow, and negative relative offsets are not allowed. As designed this should not be a problem; Since a Sequence must be loaded in memory and locked when in use, it makes sense for it to be a contiguous block. The application can then create relative sequences, relative to the first phrase. When the sequence is loaded into memory the address of the first phrase is used as the BaseAddress.

Possible Errors from the this call:

miNoBufErr = \$2007
NoStartErr = \$1A05

Errors returned to ErrHndlrRoutine:

NoRoomMidiErr = \$1A00
NoCommandErr = \$1A01
NoRoomErr = \$1A02
NoNoteErr = \$1A04 Couldn't find note for note off.
AlreadyOn = \$1924 from Note Synth.
NoneAvailable = \$1921 from Note Synth.
miToolsErr = \$2004 From Midi Tool Set

An Example

```

Sequence      DC.L FirstPhrase
              EXPORT FirstPhrase
FirstPhrase   DC.L 01 ; Mark as a phrase
              DC.L SeqAgain-FirstPhrase
              DC.L $FFFFFF ; End of FirstPhrase

SeqAgain      EXPORT SeqAgain
              DC.L 00 ; Mark as a Pattern
              DC.L chord+t5+short+note+c4+100 ; CowBell C short
              DC.L chord+t7+eigh+note+c3+105 ; Base C 8
              DC.L $FFFFFF ; End of pattern

```

CallRoutine Command

This is a new SeqItem command that allows you to have the Note Sequencer call any routine from the Sequence itself.

```

Cmd=          30
Chord=        1 {Set as desired, but there will be a slight delay if this bit isn't set.}
Val 1=        0
Note=         0
Lbyte=        low byte of address for the routine to be called.
Hbyte=        High byte of address for the routine to be called.

```

The bank for the address for the routine to be called is the same as the data bank, when the NoteSequencer is started up. The routine should return with and RTL, and it's called with the Note Sequencers DP. The Note Sequencer Restore the Data Bank, Processor Status, and DirectPage registers. The Routine is called at Interrupt time with interrupts disabled, and a very low stack, so the routines should be very careful as to how they are used, and should Poll MIDI to insure that MIDI Input isn't lost.