Cheezy Synth Sequencer Format

The idea of this format is that it needs to be simple, use a minimum of data, and very effective. The format is also tailored to the device it is going to be running on and is not intended to be a general purpose sequencer format. A sequence is built up of a series of events. Each event can be made up of as many sequencer objects as are needed.

Sequencer Objects:

|  |  |  |  |
| --- | --- | --- | --- |
| VALUE | DEFINE | BYTES | DESCRIPTION |
| 0x80 | VCO1\_PITCH | 3  0:0x80  1:<Note Value>  2:<Beats> | Set the pitch of VCO1 to the given value |
| 0x81 | VCO2­\_PITCH | 3  0:0x81  1:<Note Value>  2:<Beats> | Set the pitch of VCO2 to the given value |
| 0x82 | ADSR\_TRIG | 3  0:0x82  1:<Gate idth>  2:<Beats> | Trigger the ADSR |
| 8x83 | DRUM1\_PITCH | 4  0:0x83  1:<Trigger>  2:<pitch>  3:<beats> | Trigger Drum 1 |
| 0x84 | DRUM2\_PITCH | 4  0:0x84  1:<Trigger>  2:<Pitch>  3:<Beats> | Trigger Drum 2 |
| 0x85 | DRUM3\_PITCH | 4  0:0x85  1:<Trigger>  2:<Pitch>  3:<Beats> | Trigger Drum 3 |
| 0x86 | CYMBAL\_PITCH | 4  0:0x86  1:<Trigger>  2:<Pitch>  3:<Beats> | Trigger the cymbal. |
| 0x87 | REPEAT\_BEGIN | 1  0:0x87 | Start of repeat sequence. Repeats can be nested. |
| 0x88 | REPEAT\_END | 2  0:0x88  1:<Count> | End of a repeat sequence. Will repeat the sequence <count> times. |
| 0x89 | EVENT\_END | 1:  0:0x89 | Marks the end of an Event. |
| 0x8A | SEQ\_END | 1  0:0x8A | Marks the end of the sequence. The sequence will stop. |
| 0x8B | EVENT\_REST | 2  0:0x8B  1:<Beats> | Does Nothing, but does create a timer that will generate the beginning of an event. |

When the sequencer starts it parses the data until it reaches the end of event marker. For each object that will last for a certain number of beats, a timer is started. On each tick the timers are decremented and when a timer goes to zero the sequencer will parse more data until it reaches the next event end marker. This implies that care must be used in selecting the beats for each event so that they are all in sync, otherwise odd things will begin to happen and your sequence is not going to sound as expected.

Timers are allocated and freed by a timer manager. As soon as an event has ended, the timer associated with it is freed so it can be used again.