

# **EWMOD**

## **An Envelope and Waveform Modification Module for the Music 5000**

Angus Duggan

6 March 1989

## Introduction

This document describes a software module which enables modification of envelopes and waveforms for the Music 5000 synthesiser. The module requires the M5 module from the Music 5000 Release 2 software to be available. The EWMOD module is designed as a core module for use by envelope and waveform editors, presenting a minimal set of words for use. Conversion of parameters from easily understandable forms are described in the cases where data is required in awkward forms.

## Envelopes

Envelopes are used to modify the amplitude or pitch of notes as they play. Envelopes in the Music 5000 system consist of three sections, each of which may be as simple or complex as required. The pitch or amplitude of an envelope at any point is referred to as the *level* of the envelope. The *on* section defines what happens to the level of the envelope when a note is started. When the on section has finished, the note enters the *repeat* section, which is continually repeated until the note is turned off or on again. When the note is signalled to turn off, the envelope enters the *off* section, and when the *off* section has finished the envelope level returns to zero until the note is turned on again.

Each section of an envelope is made up of zero or more *segments*. Each segment has a *gradient* and a *target level*. The gradient defines how quickly the envelope level changes to the target level from its current level. An envelope will never overshoot the target level, and the envelope will always wait until the target level is reached before moving on to the next segment.

The M5 module provides 110 segments to be shared between 34 envelopes. There are no restrictions on the number of segments allowed in any individual envelope or section except the overall total. The envelopes are numbered from 0 to 33, and the M5 module assumes a convention by which envelopes 0–15 are pitch envelopes, envelopes 16–31 are amplitude envelopes, and envelopes 32 and 33 are default envelopes which should not be modified. It is suggested that this convention is followed, to prevent confusion between pitch and amplitude envelopes.

The words in the EWMOD module which operate on envelopes are —

### *first last Eclr*

This word clears the envelopes from *first* to *last*, freeing the segments used by them for use. Envelopes can be redefined without clearing them first, but if a large number of segments are required extra envelopes may have to be cleared to make space.

### ***on repeat off scale envnum Emake***

This word redefines the envelope specified by *envnum*. Any segments used by envelope *envnum* are freed for use, and then the number of segments specified by *on*, *repeat*, and *off* parameters are allocated to the envelope. The number of segments allocated to any section can be zero, in which case the section is ignored. If zero repeat segments are specified, the level of the envelope will remain constant at the final level of the on section until the note is turned off.

The *scale* parameter indicates whether the levels should be magnified. A negative scale indicates that no magnification should be performed, zero indicates that levels should be multiplied by 4, and a positive scale indicates that levels should be multiplied by 16. Scaling is especially useful for pitch envelopes which are required to sweep a large pitch range.

No initialisation of the segments allocated is performed, so all segments *must* be redefined for the envelope.

Note that most envelopes will have a segment at the start of the on section which resets the level to zero immediately, so that all subsequent changes can take place on a known timescale.

### ***gradient level segnum Eseg***

This word redefines the segment number *segnum*, in the envelope last initialised by **Emake**. Segments are numbered from 0 up to one less than the total number of segments in the on, repeat, and off sections. The *level* parameter specifies the target level at the end of the segment, and the *gradient* parameter specifies the rate at which to alter the level.

The level parameter is the target volume or pitch change plus 127. For amplitude envelopes, only values in the range 127–255 are meaningful, corresponding to amplitudes of 0–128. For pitch envelopes, the full range of 0–255 can be used, corresponding to –127 to +128 pitch units. The size of the pitch units depends upon the scale set by **Emake** – it will be  $\frac{1}{16}$  semitones for a negative scale,  $\frac{1}{4}$  semitones for a zero scale, or 1 semitone for a positive scale.

The gradient can be calculated by multiplying the *magnitude* of the level change desired by 256, and dividing the result by the number of time units over which the change is to take place. Each time unit is approximately 10 milliseconds. Note that the gradient is not signed; the direction in which the level will change is defined by the difference between the current level and the target level.

A gradient of zero should not be specified; if a long period with no change is desired, a level difference of 1 should be specified, with a gradient sufficiently shallow to take up the time required.

### ***envnum* Eset**

This word sets the envelope indicated by *envnum* to be the pitch or amplitude envelope for the current channel selection. The M5 words **AEN** and **PEN** should be used to distinguish whether the envelope is to be set as the amplitude or pitch envelope.

## **Waveforms**

Waveforms define the pressure fluctuations which characterise a sound. The Music 5000 synthesiser is capable of storing 14 waveforms of 128 points each, and modulating these waveforms using a variety of special effects. The waveform defining system provided by EWMOD allows more waveforms to be prepared and loaded into the Music 5000 as desired. The waveforms are numbered from 0 to 13.

The system used in EWMOD revolves around waveform *buffers*. These are 128 word arrays which are provided by the programmer (possibly by writing 128 DIM ARRAY). If waveforms are set at the start of a program and never altered afterwards, one buffer can be used to build all of the waveforms in turn. If waveforms are to be altered during execution of a program it is better to build all of the waveforms required in different buffers, and then load them from the buffers when required. Another advantage of using buffers is that words can be written using the normal load and store operators to perform filtering or other operations on partially generated waveforms. Note: the size of the buffer provided is not checked by any of these words, and so use with a wrongly sized or addressed buffer will probably corrupt the program. It is recommended that programs modifying waveforms are saved before running.

### ***buffer* Wclr**

This word resets the waveform in *buffer* to be zero at all points. This is usually necessary before using **Wharm**.

### ***size harmonic buffer* Wharm**

This word adds a harmonic component to the waveform in *buffer*. The harmonic specified by *harmonic* should be in the range 1–16. The size of the smallest component specified by *size* should be 256, where possible. **Wmax** can be used to rescale to waveform.

### ***height1 height2 length from* Wline**

This word creates a straight line segment in a waveform. The start address of the segment is passed in the parameter *from*, and the length of the segment is specified by *length*. The parameters *height1* and *height2* specify the levels at the start and

end of the line segment respectively. **Wline** allows waveforms which could not be created by **Wharm** to be created easily. Note: *from* is the segment start address, not the buffer start address.

### ***buffer wavenum Wmake***

This word loads the waveform specified by *buffer* into the Music 5000 as waveform number *wavenum*. It also performs a **Wset** to set the waveform on the current channel selection.

### ***buffer Wmax***

This word rescales the waveform stored in *buffer*, so that the largest amplitude of any point on the waveform is &7F00. This gives the largest sensitivity to amplitude changes that the waveform can have. **Wmax** is not needed in cases where the maximum amplitude is known to be &7F00 already.

### ***wavenum Wset***

This word sets the waveform indicated by *wavenum* to be the waveform used by the current channel selection.

## **Errors**

### **Invalid envelope**

This error occurs when an envelope number outside the range 0–33 is used with **Eclr**, **Emake** or **Eset**.

### **Invalid harmonic**

This error occurs when a harmonic number outside the range 1–16 is used with **Wharm**.

### **Invalid length**

This error occurs when the *length* parameter to **Wline** is outside the range 1–128.

### **Invalid segment**

This error occurs when a segment number is used which is outside the range 0 to  $x - 1$ , where  $x$  is the total number of on, repeat and off segments in the current envelope. The current envelope is the envelope specified by the last **Emake**.

## Invalid waveform

This error occurs when a waveform number outside the range 0–13 is used in **Wset** or **Wmake**.

## Too many segments

This error indicates that all of the free envelope segments have been used up. Segments can be freed by clearing unwanted envelopes with **Eclr**. If there are still not enough segments, one or more envelope definitions may have to be simplified. If envelopes are not used simultaneously they can share the same envelope number, but care must be taken to ensure that notes using the previous envelope have finished playing before re-using the envelope number.

## Examples

Here are some examples of words which use **EWMOD** – the first two words define a pitch envelope for an increasing vibrato and a soft-edged amplitude envelope.

```
"incvib" [ % increasing vibrato
  6 2 0 ON 0 Emake % no off section, normal scale
  &7F00 127 0 Eseg % pitch 0 immediately
  &0010 126 1 Eseg % pitch -1 in 16 steps
  &0030 129 2 Eseg % pitch +2 in 16 steps
  &0060 123 3 Eseg % pitch -4 in 16 steps
  &00A0 133 4 Eseg % pitch +6 in 16 steps
  &00E0 119 5 Eseg % pitch -8 in 16 steps
  &0200 135 6 Eseg % pitch +8 in 16 steps
  &0200 119 7 Eseg % pitch -8 in 16 steps
  0 PEN Eset      % make current pitch envelope
]

"soft" [ % soft attack
  3 0 2 ON 16 Emake % no repeat section, normal scale
  &7F00 127 0 Eseg % start at 0 amplitude
  &3800 239 1 Eseg % amplitude 112 in 2 steps
  &0200 255 2 Eseg % amplitude 128 in 8 steps
  &2000 158 3 Eseg % amplitude 31 in 3 steps
  &0100 127 4 Eseg % amplitude 0 in 31 steps
  16 AEN Eset     % make current amplitude envelope
]
```

The next words define a triangular wave and a “reedy” wave (waveforms with odd harmonics tend to produce sounds similar to reed instruments). Both of the words use the same waveform buffer, but separate buffers could be used if desired.

```
"buffer" [ % waveform buffer
  128 DIM ARRAY
]

"triangle" [ % triangular wave
  0 100 32 1 buffer Wline      % rise until 1/4 wave
  100 -100 64 32 buffer Wline % fall until 3/4 wave
  -100 0 32 96 buffer Wline   % rise until end of wave
  1 buffer Wmax               % scale wave
]

"reedy" [ % reed like sound
  1 buffer Wclr               % clear buffer
  100 1 1 buffer Wharm % strong first harmonic
  80 3 1 buffer Wharm % strongish third harmonic
  60 5 1 buffer Wharm % medium fifth harmonic
  40 7 1 buffer Wharm % weakish seventh harmonic
  1 buffer Wmax             % scale wave
]

"select" [ % select buffer as wave 1
  1 buffer 1 Wmake % set waveform and select
]
```