



MeloDream V1.00

2007 © von Marc-Sven Rudolf
Alle Rechte vorbehalten.

Lizenz

§1 - Verwendung

- 1.) Es ist erlaubt die Library in eigenen Projekten kostenlos zu verwenden.
Das gilt für kommerzielle sowie für freie Software.
- 2.) Die Verwendung der Library ist nur erlaubt wenn dadurch keine der Punkte dieser Lizenz-Erklärung verletzt werden.

§2 – Urheberrecht

- 1.) Das Urheberrecht dieser Library bleibt in jedem Fall bei Marc-Sven Rudolf, dem Autor dieser Library.
- 2.) Bei Verwendung dieser Library in eigener Software muss der Autor deutlich in den Credits der Software im Bezug auf diese Library genannt werden.
- 3.) Es ist nicht gestattet sich als Autor dieser Library auszugeben.

3§ - Haftung

- 1.) Die Haftung für Fehler und sonstige Mängel des Programms richtet sich nach den außerhalb dieser Lizenz getroffenen Vereinbarungen zwischen Ihnen und den Rechtsinhabern oder, wenn eine solche Vereinbarung nicht existiert, nach den gesetzlichen Regelungen.
- 2.) Für Schäden die durch kriminelle oder fahrlässige Handhabung dieser Library entstanden sind, kann keinerlei Haftung übernommen werden.
- 3.) Zuwiderhandlungen gegen diese Lizenz werden generell strafrechtlich verfolgt.

Was ist MeloDream ?

MeloDream ist eine Library welche es Programmierern ermöglicht, unter allen Programmiersprachen die DLLs verwenden können über die Angabe von Opcodes bis zu 32 stimmige Musikstücke OnTheFly zu erstellen.

Die Opcodes ähneln jenen des QuickBasic Play-Befehls und denen der Handy-Klingelton Editoren sehr.

Die Lib spielt die Musikstücke in sehr realistischer Qualität ab.

Es können sogar allerlei Effekte wie Chorus, Release, Lautstärke für einzelne Noten-Sequenzen realisiert werden.

Die Library verwendet die DirectX Software-Wavetable.

Diese wird ebenfalls von einer Vielzahl von Computerspielen zum Abspielen von Hintergrundmusik genutzt.

Systemanforderungen:

- Microsoft Windows XP/2000/VISTA
- DirectX ab 8.x
- Minimum 8 MB freier RAM
- Eine Soundkarte

Die Funktionen der Library

Funktionsdaten der 'MeloDream.dll':

void	MELODREAM_Init	(void)
long/bool	MELODREAM_SetChannel	(long Channel , char/string Opcodes)
long/bool	MELODREAM_Play	(long Speed , bool/byte Loop)
long/bool	MELODREAM_Stop	(void)
long/bool	MELODREAM_Pause	(void)
long/bool	MELODREAM_Resume	(void)
long/bool	MELODREAM_Restart	(void)
long/bool	MELODREAM_Clear	(void)
long/bool	MELODREAM_SaveSong	(char/string Filename)
long/bool	MELODREAM_LoadSong	(char/string Filename)
long/bool	MELODREAM_SetPosition	(long NoteIndex)
long	MELODREAM_GetPosition	()
long/bool	MELODREAM_SetBPM	(long BPM)
long	MELODREAM_PeekLevel	()

Funktionen im Detail:

MELODREAM_Init()

Erzeugt die notwendigen Umgebungsdaten zum Verwenden der Library.
Diese Funktion muss vor allen anderen MeloDream-Funktionen aufgerufen werden.

Diese Funktion hat keinen Rückgabewert.

```
result = MELODREAM_SetChannel (long Channel , char/string Opcodes)
```

Mit dieser Funktionen können sie Melodiedaten über Opcodes an einen Abspiel-Kanal senden.

Die Funktion hängt alle angegebenen Daten an eventuell bereits existierende Daten auf dem Kanal an.

- **Channel** muss eine Nummer zwischen 0 und 31 sein. Sie bestimmt den Abspielkanal
- **Opcodes** muss ein String mit Opcodes sein.

Mehr Informationen über die Opcodes im Kapitel „Opcodes“.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_Play (long Speed , bool/byte Loop)
```

Hiermit kann das im Speicher befindliche Musikstück abgespielt werden.

- Speed muss eine Nummer zwischen 50 und 250 sein. Sie bestimmt die Abspiel-Geschwindigkeit.
- Loop muss ein True oder ein False Wert sein. True bestimmt dass das Stück am Ende automatisch wiederholt wird.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_Stop ()
```

Hält das zur Zeit abspielende Stück an.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_Pause ()
```

Pausiert das zur Zeit abspielende Stück an.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_Resume ()
```

Setzt das zur Zeit abspielende und pausierte Stück fort.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_Restart ()
```

Startet das Abspielen des Stückes neu.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_Clear ()
```

Löscht den Songspeicher.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_SaveSong (char/string Filename)
```

Speichert den Songspeicher in eine Datei um sie zu einem andere Zeitpunkt wieder laden zu können.

- **Filename** muss ein String mit dem Dateinamen sein.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_LoadSong (char/string Filename)
```

Lädt einen Song von einem Datenträger den Songspeicher.
Der Song muss zuvor per **MELODREAM_LoadSong** abgespeichert worden sein.

- **Filename** muss ein String mit dem Dateinamen sein.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_CatchSong (long/pointer Address)
```

Speichert den Songspeicher in eine Datei um sie zu einem andere Zeitpunkt wieder laden zu können.

- **Address** muss eine Speicheradresse zu dem Song sein der geladen werden soll.
Der Song muss zuvor per **MELODREAM_LoadSong** abgespeichert worden sein.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_SetPosition (long NoteIndex)
```

Setzt eine neue Position von der aus abgespielt werden soll.
Diese Funktion kann nur während des Abspielens verwendet werden.

- **NoteIndex** muss eine Speicheradresse zu dem Song sein der geladen werden soll.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_GetPosition ()
```

Ermittelt die akutelle Abspielposition.
Diese Funktion kann nur während des Abspielens verwendet werden.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_SetBPM (long BPM)
```

Setzt eine Abspielgeschwindigkeit.
Diese Funktion kann nur während des Abspielens verwendet werden.

- **BPM** muss ein Wert von 50 bis 250 sein.

Die Funktion gibt True zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls False.

```
result = MELODREAM_PeekLevel ()
```

Ermittelt das Outputlevel der abspielenden Melodie.
Diese Funktion kann nur während des Abspielens verwendet werden.

Die Funktion gibt einen Numerischen Wert zurück wenn sie erfolgreich ausgeführt wurde.
Andernfalls Null.

Opcodes

Die Opcodes sind dazu da eine Melodie auf einem Abspielkanal zu definieren. Sie sind in der Art und in der Syntax sehr ähnlich mit vielen Klingelton-Editoren für Handies und für alle diees noch kennen, auch mit dem QuickBasic Play-Befehl.

Es gibt 3 Arten von Opcodes.

Zum einen die Effektcodes welche zum Setzen der Lautstärke, des Choruses und der Releasetime da sind.

Zum andern die Statuscodes welche globale Auswirkungen auf einem Abspielkanal haben.

So zum Beispiel für das setzen eines Instrumentes oder eines Midi-Kanals.

Dann wären da noch die Notencodes. Mit deren Hilfe lässt sich letztendlich die Melodie definieren die auf einem Abpielkanal angelegt werden soll.

Statuscodes:

Instrument für einen Kanal setzen.

i [NUMMER]

„Nummer“ muss ein Wert von 0 bis 127 sein.

Die Werte dieser Definition sind Nummer von Instrumenten.

Mehr dazu im Kapitel „Instrumente“.

Diese Definition sollte stets am Anfang einer Kanalmelodie stehen.

Die Werte dieser

Midi-Kanal wählen.

m [NUMMER]

„Nummer“ muss ein Wert von 0 bis 31 sein.

Sie bestimmt den Midi-Ausgabekanal.

Bitte beachten sie dass pro Midi-Kanal nur ein Instrument nutzbar ist.

Kanal 9 ist die Schlagzeug-Spur.

Der Midi-Kanal ist nicht mit dem Ausgabekanal zu verwechseln.

Ein Midi-Kanal kann auch gleichzeitig auf mehreren Ausgabekanälen angewendet werden.

Achtung:

Die Statuscodes wirken global auf einen Ausgabekanal.

Sie müssen also nur am Anfang einer Kanalmelodie definiert werden.

Notencodes:

Folgende Noten sind in der Library bekannt.

C D E F G A H

Code-Definition:

NOTE[#]OKTAVE

Die Raute '#' definiert, dass die Note einen Halbton höher gespeichert wird.

Die Raute ist daher optional.

Die Oktave muss stets angegeben werden.

Gültige Werte sind 0 - 9.

Beispiel (Tonleiter. Die Leerzeichen sind optional):

c2 c#2 d2 d#2 e2 f2 f#2 g2 g#2 a2 a#2 h2 c3

Pausenzeichen:

Es wird zwischen Ton haltenden und Ton endenden Pausen unterschieden.

Das Zeichen 'p' macht eine Pause von einem viertel Takt und beendet den letzten Ton.

Das Zeichen '_' macht ebenfalls eine Pause von einem viertel Takt, hält aber den letzten Ton an.

Beispiel (Takt):

c2 _ _ _ c2 p p p

Effektcodes

Lautstärke für nachfolgende Noten setzen:

v [NUMMER]

„Nummer“ muss ein Wert von 0 bis 127 sein.

Sie bestimmt die Lautstärke für alle nachfolgenden Noten.

Chorus für nachfolgende Noten setzen:

xc [NUMMER]

„Nummer“ muss ein Wert von 0 bis 127 sein.

Sie bestimmt die Chorus-Intensivität für alle nachfolgenden Noten.

xr [NUMMER]

„Nummer“ muss ein Wert von 0 bis 127 sein.

Sie bestimmt die Release-Time für alle nachfolgenden Noten.

Instumente

Nummer / Instrument

Pianos/Klaviere

0	B_ACOUSTIC_GRAND
1	B_BRIGHT_GRAND
2	B_ELECTRIC_GRAND
3	B_HONKY_TONK
4	B_ELECTRIC_PIANO_1
5	B_ELECTRIC_PIANO_2
6	B_HARPSICHORD
7	B_CLAVICHORD

Schlaginstrumente

8	B_CELESTA
9	B_GLOCKENSPIEL
10	B_MUSIC_BOX
11	B_VIBRAPHONE
12	B_MARIMBA
13	B_XYLOPHONE
14	B_TUBULAR_BELLS
15	B_DULCIMER

Orgeln

16	B_DRAWBAR_ORGAN
17	B_PERCUSSIVE_ORGAN
18	B_ROCK_ORGAN
19	B_CHURCH_ORGAN
20	B_REED_ORGAN
21	B_ACCORDION
22	B_HARMONICA
23	B_TANGO_ACCORDION

Gitarren

24	B_ACOUSTIC_GUITAR_NYLON
25	B_ACOUSTIC_GUITAR_STEEL
26	B_ELECTRIC_GUITAR_JAZZ
27	B_ELECTRIC_GUITAR_CLEAN
28	B_ELECTRIC_GUITAR_MUTED
29	B_OVERDRIVEN_GUITAR
30	B_DISTORTION_GUITAR
31	B_GUITAR_HARMONICS

Bässe

32	B_ACOUSTIC_BASS
33	B_ELECTRIC_BASS_FINGER
34	B_ELECTRIC_BASS_PICK
35	B_FRETLESS_BASS
36	B_SLAP_BASS_1
37	B_SLAP_BASS_2
38	B_SYNTH_BASS_1
39	B_SYNTH_BASS_2

Streicher und Timbanies

40 B_VIOLIN
41 B_VIOLA
42 B_CELLO
43 B_CONTRABASS
44 B_TREMOLO_STRINGS
45 B_PIZZICATO_STRINGS
46 B_ORCHESTRAL_STRINGS
47 B_TIMPANI

Streicher und Stimmen

48 B_STRING_ENSEMBLE_1
49 B_STRING_ENSEMBLE_2
50 B_SYNTH_STRINGS_1
51 B_SYNTH_STRINGS_2
52 B_VOICE_AAH
53 B_VOICE_OOH
54 B_SYNTH_VOICE
55 B_ORCHESTRA_HIT

Bläser

56 B_TRUMPET
57 B_TROMBONE
58 B_TUBA
59 B_MUTED_TRUMPET
60 B_FRENCH_HORN
61 B_BRASS_SECTION
62 B_SYNTH_BRASS_1
63 B_SYNTH_BRASS_2
64 B_SOPRANO_SAX
65 B_ALTO_SAX
66 B_TENOR_SAX
67 B_BARITONE_SAX
68 B_OBOE
69 B_ENGLISH_HORN
70 B_BASSOON
71 B_CLARINET

Flöten/Pfeifen

72 B_PICCOLO
73 B_FLUTE
74 B_RECORDER
75 B_PAN_FLUTE
76 B_BLOWN_BOTTLE
77 B_SHAKUHACHI
78 B_WHISTLE
79 B_OCARINA

Synthesizer

80	B_LEAD_1	B_SQUARE_WAVE
81	B_LEAD_2	B_SAWTOOTH_WAVE
82	B_LEAD_3	B_CALLIOPE
83	B_LEAD_4	B_CHIFF
84	B_LEAD_5	B_CHARANG
85	B_LEAD_6	B_VOICE
86	B_LEAD_7	B_FIFTHS
87	B_LEAD_8	B_BASS_LEAD
88	B_PAD_1	B_NEW_AGE
89	B_PAD_2	B_WARM
90	B_PAD_3	B_POLYSYNTH
91	B_PAD_4	B_CHOIR
92	B_PAD_5	B_BOWED
93	B_PAD_6	B_METALLIC
94	B_PAD_7	B_HALO
95	B_PAD_8	B_SWEEP

Synthesizer Effekte

96	B_FX_1
97	B_FX_2
98	B_FX_3
99	B_FX_4
100	B_FX_5
101	B_FX_6
102	B_FX_7
103	B_FX_8

Ethnisches

104	B_SITAR
105	B_BANJO
106	B_SHAMISEN
107	B_KOTO
108	B_KALIMBA
109	B_BAGPIPE
110	B_FIDDLE
111	B_SHANAI

Percussion

112	B_TINKLE_BELL
113	B_AGOGO
114	B_STEEL_DRUMS
115	B_WOODBLOCK
116	B_TAIKO_DRUMS
117	B_MELODIC_TOM
118	B_SYNTH_DRUM
119	B_REVERSE_CYMBAL

Sound Effekte

120	B_FRET_NOISE
121	B_BREATH_NOISE
122	B_SEASHORE
123	B_BIRD_TWEET
124	B_TELEPHONE
125	B_HELICOPTER
126	B_APPLAUSE
127	B_GUNSHOT