

Extreme Midi Mixer

Autor programu: Włodzimierz Grabowski

Opis programu

Extreme Midi Mixer to program pozwalający na zbudowanie własnego, graficznego edytora komunikatów MIDI. Za jego pomocą wykorzystując porty MIDI możemy sterować dowolnym urządzeniem lub oprogramowaniem.

Podczas pierwszego uruchomienia program domyślnie wczytuje plik skryptu "**def.mixer**". Plik ten jest podzielony na Strony. Strona służy do konstrukcji szablonu danego obiektu z którego później korzystają zdefiniowane elementy (fader, knob, button, x-y, bender).

Formaty komunikatów

System Exclusiv

Przykład:

```
#8 fader 100 200 "OPIS" sysex F0 43 10 4B 50 0 17 10:20 F7
```

Taki zapis definiuje element suwakowy (fader nr 8) który będzie umieszczony na ekranie miksera o współrzędnych ekranowych $x=100$; $y=200$. Numeracja każdego elementu za pomocą znaku (hash) '#' służy do identyfikacji danego elementu podczas wczytywania i zapisywania ustawień za pomocą migawki. Współrzędne definiujemy dla lewego górnego rogu elementu, wyjątek stanowią pokręta obrotowe, gdzie definiujemy położenie środka pokręta.

Grupa sysex definiuje dane MIDI jakie zostaną wysłane podczas poruszania tym suwakiem na ekranie. Dane dla komunikatów Sysex są w formie szesnastkowej (HEX) w zakresie od 00:7F. 5. Zamiast wartości szesnastkowych (HEX) można używać także liczb dziesiętnych. W tym celu przed liczbą należy wstawić przedrostek 'i' np. **F0 i16 i32 i48 i64 F7** co będzie równoznaczne **F0 10 20 30 40 F7** (H).

Istnieje możliwość podpięcia wielu komunikatów sysex i control pod jeden element

Przykład:

```
#1 fader 10 10 "SYSEX COMBI"
```

```
sysex F0 1A 1B 0 F7
```

```
sysex F0 1A 1B 1 F7
```

```
sysex F0 1A 1B 2 F7
```

Opcje dodatkowe:

i - komunikaty SysEx mają format szesnastkowy (hex) i podawanie wartości domyślnych musi być również w formacie szesnastkowym. Za pomocą przedrostka 'i' możemy podawać teraz te wartości w formie dziesiętnej np. **default i100** zamiast formy szesnastkowej (**default 64**). Zasady te obowiązują tylko w przypadku SysEx. Dla danych **ctrl** wartości domyślne są zawsze w formacie dziesiętnym. Wartości domyślne są przywracane po użyciu resetu migawki.

korg- polecenie to wstawione przed zakresem wartości, będzie zamieniać je na 2-bajtowy format 14-bitowych danych firmy Korg

nibblex - opcja dzieląca bajt na tzw. pół-bajty, **x** oznacza ilość poprzedzających bajtów które mają ulec podzieleniu

Przykład:

```
#1 fader 510 790 "Harmony" sysex F0 41 00 00 23 12 20 02 22 nibble2 0:21 rolandcrc5 F7
```

rolandcrcx - funkcja licząca sumę kontrolną **crc** dla urządzeń Roland, **x** oznacza ilość poprzedzających bajtów z których ma zostać wyliczona suma kontrolna

Przykład:

```
#1 fader 1010 790 "Attack" sysex F0 41 00 00 23 12 20 02 1A nibble2 0:65 rolandcrc5 F7
```

twobyte - polecenie w definicji danych SysEx lub ctrl, rozбивa wartość spoza zakresu $0 \div 127$ na dwa bajty. Jest to przydatne np. dla komunikatów **pitch bend** lub innych danych wymagających rozdzielenia na **LSB** i **MSB**

Control

Przykład:

```
#801 fader "VOL1" ctrl B0 7 0:127
```

Dane dla komunikatów Control wprowadzamy w formie dziesiętnej (zakres $0 \div 127$), za wyjątkiem pierwszego bajtu statusu, który jest w formacie szesnastkowym (HEX) definiującym rodzaj kontrolera oraz numer kanału MIDI (np. B0:BF w przypadku Control Change). Dwukropek zwykle definiuje zakres danych (min/max) przypisanych do ruchu danego elementu (np. suwaka lub pokręta).

OBIEKTY

background: - definiuje teksturę tła, rozmiary okna i rozmieszcza dodatkowe bitmapy

fader: - definiuje element suwakowy czyli ekranowy fader

Dostępne funkcje:



autoplace

- automatyczne rozmieszczanie elementów na planszy.

Przykład:

```
autoplace 100 100 0 20
#801 fader "VOL1" ctrl B0 7 0:127
#802 fader "VOL2" ctrl B1 7 0:127
#803 fader "VOL3" ctrl B2 7 0:127
```

barbitmap

- element słupkowy dla suwaków. Np. *barbitmap "slupek.bmp"*

bendermode

- tryb powracania do ustawienia domyślnego po zwolnieniu suwaka lub gałki

Przykład:

```
fader:
bendermode
bitmap "suwaczek.bmp"
vbitmap "suwaczek_add.bmp" 16 13
move vert 128
text white bottom center fixedwidth box ltgreen
vtext 0 22 bottom center fixedwidth bold white box dkgreen
stretch 2 1
#1200 fader 545 255 "PITCH|BENDER" ctrl E0 twobyte 0:16384 defaultposition 0.5
```

default

- definiowanie domyślnego położenia suwaków podczas wyczytywania miksera, gałek. Np. *default 127*

fixedposition

- ta opcja w definicji elementu powoduje, że dany element suwak, button, przycisk będzie miał zawsze stałą wartość w zakresie od 0.0 do 1.0. Np. *fixedposition 0.5* spowoduje to, że pozycja elementu będzie zawsze w centrum. W przypadku przycisków definicja *fixedposition 1.0* spowoduje stałe wciśnięcie

hidemouse

- ukrywanie myszki podczas regulowania

hoverbitmap

- podświetlanie dla obiektów suwakowych fader i przycisków button

inverse

- ta opcja w definicji elementów button, knob lub fader powoduje, że wartości zmieniają się odwrotnie. Ten sam efekt można uzyskać również poprzez odwrotną notację zakresu danych do sterowania. Np. *i100:50* powoduje inwers czyli wartości będą zmieniać się malejąco. Przykładowe zastosowanie: *button inverse, knob inverse, fader inverse*

move horiz [zakres w pikselach]

- zakres poziomego ruchu suwaka. Np. *move horiz 100* oznacza że, ruch suwaka czyli bitmapy zmiennej **vbitmap** po ekranie będzie odbywał się w zakresie 100 pikseli od lewej do prawej

rows [ilość wierszy]

- opcja listy rozwijalnej **list**: i elementów **list** i **selectlist**, umożliwia podział listy na kolumny

step

- parametr określający co ile zmieniać się będą wartości podczas regulacji

table

- tekstowe opisywanie wartości.

Przykład:

```
table 0 "min" 64 "centrum" 127 "max"
```

Druga komenda **table2** zrobi to samo lecz dla drugiego zdefiniowanego tekstu zmiennego **vtext**. Wykorzystując **table** i **table2** możemy np. opisać dwie osie dla kontrolera X-Y. Polecenie to należy wstawić za poleceniem definicji elementu knob, button, fader, gdyż dla każdego elementu możemy ułożyć różne tabele opisowe. Opisywane wartości trzeba podawać w notacji dziesiętnej

knob: - definiuje element obrotowy knob czyli ekranową "gałkę"



Dostępne funkcje:

bendermode

- tryb powracania do ustawienia domyślnego po zwolnieniu gałki

default

- definiowanie domyślnego położenia gałek podczas wyczytywania miksera. Np. **default 127**

defaultposition

- ustawia suwak lub gałkę w zadanym procentowo położeniu domyślnym w zakresie od 0.0 do 1.0. Np. **defaultposition 0.5** ustawi suwak domyślnie w położeniu środkowym

encoder

- użycie powoduje, że knob zadziała jak kółko obrotowe „dial”. Ruch kółka zawsze jest obrotowy, więc nie ma potrzeby definiowania ruchu przez **move circ**

fixedposition

- powoduje, że dany element będzie miał zawsze stałą wartość w zakresie od 0.0 do 1.0. Np. **fixedposition 0.5** spowoduje, że pozycja elementu będzie zawsze w centrum, w przypadku przycisków definicja **fixedposition 1.0** powoduje ich stałe wciśnięcie

hidemouse

- ukrywanie myszki na obiekcie podczas zmiany parametrów

inverse

- powoduje, że wartości zmieniają się odwrotnie. Ten sam efekt można uzyskać również poprzez odwrotną notację zakresu danych do sterowania, np. **i100:50** powoduje inwers czyli wartości będą zmieniać się malejąco. Przykładowe zastosowanie:

knob inverse

move circ

- ruch kołowy dla gałek z definicją kąta, np. **move circ 45 335**

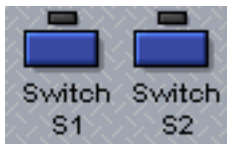
step

- określa o jaką stałą liczbę zmieniać się będą wartości gdy kręcimy kółkiem

table

- tekstowe opisywanie wartości. Np. **table 0 "min" 64 "centrum" 127 "max"**. Druga komenda **table2** robi to samo ale dla drugiego zdefiniowanego tekstu zmiennego **vtext**. Wykorzystując **table** i **table2** możemy opisać np. dwie osie dla kontrolera X-Y. Polecenie to należy wstawić za poleceniem definicji elementu knob gdyż dla każdego elementu możemy ułożyć różne tabele opisowe. Opisywane wartości trzeba podawać w notacji dziesiętnej.

button: - definiuje element przełącznikowy czyli ekranowy "przycisk"



Dostępne funkcje:

default

- definiowanie domyślnego położenia przycisku podczas wyczytywania miksera. Np. **default 127**

fixedposition

- powoduje, że dany element przycisk będzie miał zawsze stałą wartość w zakresie od 0.0 do 1.0. Np.

fixedposition 1.0 powoduje stałe wciśnięcie

group

- czasami zachodzi konieczność stosowania opcji "przełącznikowych". W skrypcie uzyskuje się to za pomocą "grupowania" komunikatów.

Przykład:

```
#302 button 224 250 "Part1" sysex F0 43 10 4B 50 0 17 1 F7 group 1
```

```
#303 button 306 250 "Part2" sysex F0 43 10 4B 50 0 17 2 F7 group 1
```

Grupując je w ten sposób spowodujemy, że wysyłany będzie zamiennie albo pierwszy, albo drugi komunikat przy czym stan pozostałych przycisków zmieni się w całej grupie. Czyli gdy naciśniemy któryś z przycisków należących do danej grupy reszta przechodzi w stan wyłączenia. Numeracja grup jest dowolna, (1, 2, ..., itd.). Grupy i stan przełączników są odpowiednio inaczej zapisywane w migawce aby wiadomo było, który z przycisków w grupie był ostatnio aktywny

hidemouse

- ukrywanie wskaźnika myszy, Istnieje możliwość zmiany parametrów za pomocą skrolowania kółkiem myszy, z wciśniętym klawiszem **Shift** zwiększamy bądź zmniejszamy wartości co 10 jednostek. Cursor musi znajdować się w polu danego elementu aby można było nim sterować klawiaturą lub scroll-em myszy. Program ma także możliwość sterowania obiektami za pomocą klawiatury (strzałki, klawisze +/- oraz **Page Down/Up**). Po najechnaniu kursorem myszy na dany obiekt strzałkami lub klawiszami możemy zmieniać wartości w przedziale +1/-1, w przypadku klawisza **Page** o +10/-10.

hoverbitmap

- podświetlanie obiektów

inverse

- powoduje, że wartości zmieniają się odwrotnie, ten sam efekt można uzyskać również, poprzez odwrotną notację zakresu danych do sterowania np. **i100:50** powoduje inwers czyli wartości będą zmieniać się malejąco. Np. **button inverse**

onestate

- wyłączenie trybu wciśnięcia dla przycisku (buttona)

plane

- planszowy przycisk ekranowy. Np. *plane 3 2* określa ilość pól na planszy (w sumie 6). Dla każdego pola możemy podpiąć osobny komunikat SysEx lub Control Change. Za pomocą planszy istnieje możliwość sterowania kanałem MIDI, w definicji elementu button należy dodać sekwencję [CHN]. Np. *#1 button 613 186 [CHN] fixedposition 1.0*, opcja *fixedposition 1.0* powoduje, że przycisk pozostanie zawsze wciśnięty.

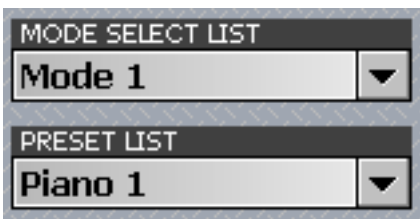
refreshbackground

- powoduje, że przed narysowaniem przycisku, odświeżane jest tło znajdujące się pod przyciskiem. Opcja niezbędna dla przycisków "transparentnych", choć zależy to od kształtu i celu ruchu przycisku. Przykładowy trójkątny przycisk **GM Reset** demonstruje zastosowanie tej funkcji.

table

- tekstowe opisywanie wartości za pomocą polecenia. Np. *table 0 "min" 64 "centrum" 127 "max"*. Druga komenda **table2** powoduje to samo ale dla drugiego zdefiniowanego tekstu zmiennego **vtext**. Wykorzystując **table** i **table2** możemy opisać dwie osie dla kontrolera X-Y. Polecenie to należy wstawić za definicją buttona i fadera gdyż dla każdego elementu możemy użyć różne tabele opisowe. Opisywane wartości trzeba podawać w notacji dziesiętnej

list: - lista rozwijalna



Dostępne funkcje:

list

- lista rozwijalna dla podanego zakresu wartości

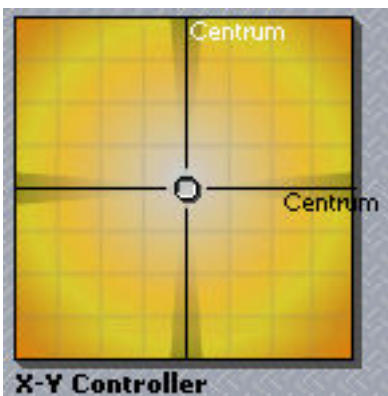
rows [ilość wierszy]

- w definicji obiektu list: umożliwia podział listy na kolumny

selectlist

- lista rozwijalna o skończonej ilości elementów. Jedna pozycja steruje tylko jednym komunikatem SysEx lub Control

xycontrol: - drążek X-Y, obiekt pracujący w trybie "2D" pozwalający sterować dwoma kontrolerami jednocześnie w różnych płaszczyznach



Dostępne funkcje:

area

- tym poleceniem ustawiamy granice elementu ruchomego **area x y** (szer; wysokość) np. *area 0 0 128 128*, w ten sposób mamy możliwość dostosowania zakresu ruchu wskaźnika do wielkości bitmapy

bendermode

- tym poleceniem spowodujemy, że kontroler będzie wracał do położenia domyślnego x, y. Działanie podobne do Pitch Bendera

step

- określa o jaką stałą liczbę zmieniać się będą wartości gdy przesuwamy element ruchomy na planszy X-Y

xaxis i yaxis

- tymi funkcjami decydujemy jakiego typu kontrolery i na której osi będą sterowane

scroll: - obiekt ten umożliwi wstawienie suwaków do przesuwania zawartości okna.

multicontrol: funkcja ta umożliwi sterowanie wieloma zmiennymi w jednym komunikacie SysEx. Można kontrolować dwa komunikaty a nawet całą serię komunikatów sysex jednocześnie w taki oto sposób:

Przykład:

```
multicontrol {
  sysex F0 43 10 0:7F 0:7F 0:7F 0:7F F7
  sysex F0 43 10 0:7F 0:7F F7

  #1 fader "A1" // ten steruje pierwszą zmienną
  #2 fader "A2" // ten steruje drugą zmienną
  #3 fader "A3"
  #4 fader "A4"
  #5 fader "B1"
  #6 fader "B2" -> ten steruje szóstą zmienną
}
```

Zmienne czyli zakres zmieniających się wartości faderów identyfikowane są dwukropkiem ':

Zmieniając cokolwiek faderami wysyłana będzie cała paczka tych komunikatów znajdująca się w zakresie opcji multicontrol. W powyższym przykładzie zostaną wysłane od razu dwa sysex-y.

Jeśli rozdzielimy sysex-y na dwa komunikaty multicontrol wtedy ruch suwaków A wysyła tylko jeden sysex ze swojej grupy, ruch suwaków B wysyła też jeden sysex oczywiście tylko ze swojej grupy multicontrol.

Przykład:

```
multicontrol {  
  sysex F0 43 10 0:7F 0:7F 0:7F 0:7F F7  
  
  #1 fader "A1"  
  #2 fader "A2"  
  #3 fader "A3"  
  #4 fader "A4"  
}
```

```
multicontrol {  
  sysex F0 43 10 0:7F 0:7F F7  
  
  #5 fader "B1"  
  #6 fader "B2"  
}
```

Można tworzyć nawet "egzotyczne" kombinacje

Przykład:

```
multicontrol {  
  sysex F0 43 0:7F 0:7F 0:7F 0:7F F7  
  sysex F0 43 0:7F 0:7F 0:7F 0:7F F7  
  sysex F0 43 0:7F 0:7F 0:7F 0:7F F7  
  ->... definicja faderów ...  
}  
multicontrol {  
  sysex F0 43 0:7F 0:7F F7  
  sysex F0 43 0:7F 0:7F F7  
  ->... definicja faderów ...  
}
```

To jaki suwak czym steruje trzeba sobie już wyliczyć. W powyższym rozbudowanym przypadku mamy 12 zmiennych w pierwszym multicontrol i tyle też należy wstawić suwaków lub gałek aby je kontrolować. W drugim multicontrol mamy 4 zmienne i aby je kontrolować trzeba wstawić cztery suwaczki albo gałki. Multicontrol koniecznie musi być zamknięty nawiasem.

Opisy tekstowe

Dostępne funkcje:

commenttime

- umożliwia ustawienie czasu trwania podpowiedzi „dymka”, np. *commenttime 3000*

fixedwidth

- umożliwia ustawienie stałej szerokości opisu text lub tekstu zmiennego **vtext**, np. *fixedwidth 100*

text

- opis obiektu graficznego

titlebar

- ustawia własny tytuł okna, jeśli nie zadeklarujemy tej funkcji na pasku programu pokaże się oryginalna nazwa programu

vtext

- tekst ukazujący się przy zmianie parametrów. Istnieje możliwość wstawienia drugiego tekstu zmiennego **vtext**, pierwszy pobiera wartości z przedostatniego komunikatu **ctrl**, drugi **vtext** pobiera wartość z ostatniego komunikatu **ctrl**

Kolor tekstu możemy definiować podając nazwę użytego koloru:

green - zielony

blue - niebieski

black - czarny

yellow - żółty

gray - szary

lime - limetka

white - biały
orange - pomarańczowy

Dodatkowo możemy stopniować nasycenie koloru przedrostkami:

lt - light (jasny); **np. ltred** - jasny czerwony
med - medium (średni); **np. medred** - średni czerwony
dk - dark (ciemny); **np. dkred** - ciemny czerwony

Dokładniejszą metodą definicji koloru jest podanie trzech składowych R,G, B (podobnie jak w HTML) notacją szesnastkową (hex). Np. **color FF8000** oznacza składowe R (czerwony) = 255, G (zielony) = 128, B (niebieski) = 0. Finalnie otrzymamy pomarańcz.

Kolorować możemy tekst oraz ramkę tekstu. Np. **text red box green** uzyskamy czerwony tekst na zielonej ramce, taki sam efekt również uzyskamy stosując polecenie **color np. text color FF0000 box color 00FF00**.

Przykład:

```
text bottom center      box orange      font "Tahoma" 8 black bold
vtext 0 12 bottom center font "Tahoma" 8 black bold
```

Grafika - elementy i ich rozmieszczanie

W grafikach (bitmapach) kolorem transparentnym (przeźroczystym) jest kolor różowy RGB (255, 0, 255). Jest on używany do definiowania pól, które nie będą rysowane (widoczne) na planszy programu. Np. do tworzenia przycisków o innym kształcie niż prostokąt. Klik na kolorze transparentnym nie powoduje zmiany stanu przycisku.

Dostępne funkcje:

ofs [x] [y]

- to polecenie przesuwa całe grupy elementów na ekranie. Jeżeli wcześniej ułożyliśmy sobie w jakimś miejscu sporo elementów to aby ponownie nie przeliczać ich pozycji ekranowych można posłużyć się opcją **ofs** podając w pikselach relatywne przesunięcie np. **ofs 50 60** przesuwa wszystkie następne elementy o 50 pikseli w poziomie i 60 pikseli w pionie aż do zakończenia tego polecenia słowem **ofs**, samo polecenie **ofs** bez parametrów dezaktywuje opcję i nie powoduje przesunięcia.

stretch

- zmienia rozmiary bitmapy, powiększając lub zmniejszając ją o zadaną wartość, czyli za jego pomocą można "rozciągnąć" obrazek (bitmapę) o zadaną wielokrotność

Przykład:

```
stretch 1 1 - nie zmienia rozmiaru bitmapy
stretch 3 0.25 - powiększy szerokość 3-krotnie, a zmniejszy wysokość 4-krotnie
```

transparentclick

- aktywuje kliknięcie na transparentnym tle, czasami jest potrzebne.

workarea [szer] [wys]

- pozwala na zdefiniowanie dowolnego obszaru roboczego (nawet kilka tysięcy pikseli, należy się jednak liczyć z obciążeniem pamięci RAM). Ta funkcja przydaje się do wstawionego obiektu scroll: gdy musimy przesuwać po dużym ekranie panelu. Poleceniem **workarea** definiujemy wielkość okna roboczego: np. **workarea 1000 1000**. Można swobodnie kombinować z różnymi ustawieniami dla obszaru roboczego i wielkości okienka aplikacji Windows uzyskując w ten sposób różne funkcjonalne cechy. 10000 x 10000 pikseli jest maksymalnym obszarem roboczym programu.

MENU programu

Mikser

Otwórz... (Ctrl+O)

- wczytuje skrypt miksera, przy pierwszym otwarciu programu będzie to plik „def.mixer”. Przy zamykaniu programu zostaje zapamiętany ostatnio wczytany plik miksera

Przeglądaj (F5)

- ponowne wczytanie załadowanego już pliku miksera (skrót klawiszowy: **F5**)

Zawsze pełny ekran przy starcie

- zaznaczenie tej opcji spowoduje wyświetlenie pełnego okna po starcie programu

Port MIDI IN wejściowy

- ustawiania portu wejściowego dla przychodzących komunikatów MIDI

Port MIDI OUT wyjściowy

- ustawianie portu wyjściowego dla komunikatów MIDI wychodzących z miksera

Statystyka

- podaje dokładne informacje o wykorzystanych przez program zasobach (m.in. ilość zajmowanej pamięci RAM, liczbę zdefiniowanych obiektów, elementów i Stron)

Debug

- po zaznaczeniu tej opcji program zapisuje i zestawia wszelkie wysłane/przetwarzane bajty w pliku tekstowym **debug.txt**, opcja bardzo przydatna w diagnostyce oraz analizowaniu działania samej aplikacji

MIDI IN Monitor (F4)

- włącznie monitora zdarzeń **MIDI Monitor**

Monitor MIDI to narzędzie do monitorowania komunikatów MIDI na wejściu MIDI In programu. Monitor ma możliwość wybrania trybu kolorowania grup przychodzących komunikatów. Można zmieniać wysokość okna podglądu zdarzeń, kopiować dane do schowka (Ctrl+C) oraz zaznaczać wszystkie wpisy (Ctrl+A). Sposób wyświetlania komunikatów można zmieniać za pomocą ikonki na górnym pasku monitora. Prawy klik na oknie podglądu wywołuje dodatkowe menu kontekstowe. Wartości kontrolerów, nut itp. są podawane w formie dziesiętnej, SysEx zawsze jest w formie szesnastkowej (HEX). Napływanie lub wysyłanie danych MIDI jest sygnalizowane przez zapalenie się lampek ikonki **MIDI** na pasku tray. Odczyt informacji następuje w formacie [sekunda].[milisekundy],[setne części milisekundy] w otrzymanym "przerwaniu IRQ" z portu MIDI, a do mierzenia czasu użyty jest zegar wysokiej rozdzielczości mierzący czas z dokładnością nanosekundową. Fakt ten można wykorzystać do celów diagnostycznych i pomiarowych.

Klawiszem **Tab** można przełączać widok pomiędzy oknem MIDI Monitora a oknem Panelu.

Wyjście

- zamykanie programu

Migawka

Migawki służą do zapamiętywania ustawień chwilowych. W projekcie mamy ich do dyspozycji 10. Posiadają skróty klawiszowe do szybkiego przywołania zawartych w nich ustawień [1] ÷ [10], z wciśniętym klawiszem [CTRL] zapisujemy ustawienia. Przywołanie migawki wysyła na port MIDI dane tylko z aktywnej **Strony**. Chcąc wysłać wszystko należy posłużyć się poleceniem **Total Dump** z menu **Eksport** programu.

Otwórz

- otwiera plik z migawkami „x.snap”

Zapisz

- zapisuje migawki do pliku

Wczytaj 1, 2,... 10

- wczytanie ustawień danej migawki

Zapisz

- wykonuje migawkę

Losuj [F8]

- wybranie tego polecenia spowoduje ustawienie losowych wartości dla wszystkich zmiennych, taki stan można zapisać pod postacią migawki

Zeruj

- wybranie tego polecenia spowoduje, iż wszystkie elementy zostaną wyzerowane do wartości „0”

Reset

- wybranie tego polecenia przywraca ustawienia domyślne z pliku startowego migawki

Export

Midi File

- zapis aktualnych ustawień elementów do pliku MIDI File

Total Dump

- wysyła wszystkie ustawienia ze wszystkich Stron na port MIDI

Page Dump

- wysyła ustawienia tylko aktywnej Strony

Kanał Midi - wybór kanału MIDI, spowoduje to zmianę zmiennej globalnej [CHN]

Strona - nawigacja po Stronach projektu

Główne cechy programu:

- sterowanie instrumentami i aplikacjami MIDI za pomocą komunikatów System Exclusive i Control Change.
- możliwość tworzenia własnego dowolnie wyglądającego graficznie mikserka.
- możliwość definiowania jedno i dwuwymiarowych elementów kontrolnych
- możliwość sterowania elementami za pomocą klawiatury i myszki (wraz z jej scrollem)
- możliwość definiowania nietypowych zmiennych (np."14-bit Korg", "Roland nibble")

- zrzut i odczyt ustawień miksera "Snap" do pamięci (10 komórek)
- pamiętanie ostatnio otwartego folderu
- pamiętanie ostatnio wczytanego pliku miksera
- tworzenie stron **Page** np. page „Performance”
- główne ustawienia programu zapisywane są w pliku konfiguracyjnym (exmixer.ini). W tym pliku zapisywany jest także odstęp czasowy pomiędzy wysyłanymi komunikatami (standardowo 5 ms)
- zapis i odczyt zrzutów ustawień „Snap” miksera na dysk
- export ustawień miksera do pliku MIDI
- zintegrowany MIDI Monitor
- polska i angielska wersja językowa
- bezpłatny do użytku domowego i komercyjnego

Instalacja:

Program nie wymaga instalowania, wystarczy go skopiować (z podkatalogami) do dowolnego katalogu, na dowolnym dysku i uruchomić plik **Exmixer.exe**.

SKRYPT - plik def.mixer

Plik **def.mixer** to zwykły plik tekstowy. Można go edytować posługując się Notatnikiem systemowym a na koniec zapisać z rozszerzeniem **.mixer**.

Struktura obiektów:

Najpierw definiujemy **OBIEKT**, a później to, do czego podpinamy komunikaty (np. SysEx), czyli **ELEMENT**. **ELEMENT** jest jakby "pochodną" obiektu, gdyż wykorzystuje jego zdefiniowane cechy. Czyli **OBIEKT** składa się z **ELEMENT**-ów, a każdy **ELEMENT** może sterować nawet kilkoma komunikatami (SysEx i CTRL).

OBIEKT definiujemy słowami kluczowymi:

fader:
knob:
button:
xycontrol:
list:

(kluczową rolę pełni tu dwukropek na końcu nazwy OBIEKT-u).

ELEMENT definiujemy w nast. sposób:

```
#[nr] fader [x] [y] "OPIS"
#[nr] knob [x] [y] "OPIS"
#[nr] button [x] [y] "OPIS"
#[nr] xycontrol [x] [y] "OPIS"
#[nr] list [x] [y] "OPIS"
```

Kluczową rolę pełni tu hash (#) i nazwa elementu (już bez dwukropka). Niektóre elementy mogą mieć funkcje specjalne jak: bendermode, onestate, encoder, transparentclick, plane itp.

Struktura:

```
OBIEKT - np. (fader:, knob:, button:, xycontrol:, list:)
  ELEMENT - np. (fader, knob, button, xycontrol, list)
    SYSEX (dane)
    ...
    CTRL (dane)
    ...
OBIEKT
  ELEMENT
    SYSEX (dane)
OBIEKT
```


ELEMENT

CTRL (dane)

itd.

Piszemy skrypt

1. Na początku skryptu umieszczamy funkcje i opcje globalne dla całego miksera:
autoload (automatyczne przywoływanie migawki z pliku „x.snap” po wczytaniu miksera)
bitmapdir (relatywny katalog gdzie znajdują się bitmapy)

Przykład:

```
autoload "start.snap"  
bitmapdir "skin"
```

2. Definiujemy obiekt **scroll**: (pasek przewijania i jego elementy graficzne):

```
scroll:  
bitmap "NAZWA_PLIKU.BMP" [SZER. STRZAŁEK]  
vbitmap "NAZWA_PLIKU.BMP"  
hoverbitmap "NAZWA_PLIKU.BMP"
```

Przykład:

```
scroll:  
bitmap "scroll.bmp" 16  
vbitmap "scroll_move.bmp"  
hoverbitmap "scroll_podswietlenie.bmp"
```

3. Definiujemy pierwszą **Stronę**

```
page [NR STRONY] "OPIS"  
window [SZER] [WYS] - okno strony  
workarea [SZER] [WYS] - obszar roboczy Strony, gdy będzie on większy niż okno Strony wtedy aktywują się scrolle, o ile wcześniej zostały zdefiniowane
```

Przykład:

```
page 1 "GM Panel Testowy"  
window 600 400  
workarea 2000 2000
```

4. Definicja tekstur i tła pierwszej Strony (BACKGROUND)

```
background:  
texture "NAZWA_PLIKU_TEKSTURY.BMP" - zamiast tekstury można zastosować czysty kolor, np.: color C0C0C0  
bitmap [x] [y] "NAZWA_PLIKU.BMP" - brak współrzędnych [x] [y] osadzi bitmapę od początku strony
```

Przykład:

```
background:  
texture "tlo.bmp"  
bitmap "obrazek1.bmp" 13 416  
bitmap "obrazek2.bmp" 60 416  
bitmap "ramka.bmp" 341 2
```

5. Obiekty i elementy

OBIEKT - np. (fader:, knob:, button:, xycontrol:, list:)
ELEMENT - np. (fader, knob, button, xycontrol, list)

Przykład:

```
fader:  
bitmap "suwaczek.bmp"  
vbitmap "suwaczek_add.bmp" 16 13  
move vert 128  
vtext top box ltgray center font "Tahoma" 10 black bold  
text bottom center box color 0030AE font "Tahoma" 10 white bold
```

stretch 1 1

| | | | |
|------------------------|------------|-------|---------|
| #1 fader 20 20 "NOTE" | ctrl 90 60 | 0:127 | |
| #2 fader 61 20 "VOL" | ctrl B0 7 | 0:127 | |
| #3 fader 102 20 "PAN" | ctrl B0 10 | 0:127 | |
| #4 fader 143 20 "EXPR" | ctrl B0 11 | 10:90 | step 10 |

6. Definicja drugiej **Strony**
7. Definicja tekstur i tła drugiej **Strony**
8. Obiekty i elementy drugiej **Strony**
9. ...

W ten sposób możemy zadeklarować maksymalnie 20 Stron. W każdej stronie możemy definiować dowolną ilość obiektów i osadzać dowolną ilość elementów a pod każdy nich podpinąć dowolną ilość komunikatów którymi ma sterować.

Zestawienie wszystkich słów kluczowych:

#

[chn] - zmienna globalna kanału MIDI

area - w obiekcie **xycontrol**: definiuje obszar po którym możemy się poruszać

autoload - automatyczne ładowanie pliku migawki

autoplace - automatyczne rozmieszczanie elementów na planszy miksera

autoplaceoff - wyłącza automatyczne rozmieszczanie elementów na planszy miksera

background: - obiekt tła

barbitmap - bitmapa poruszającego się suwaka (fadera)

bendermode - powoduje powrót suwaka lub kółka do ustawienia początkowego

bitmap - bitmapa tła po którym porusza się suwak (fader)

bitmapdir - deklaruje katalog w którym znajdują się bitmapy

box - „pudełko” tekstu opisu

button - element przycisk

button: - obiekt przycisk

color - kolor tekstu

comment - komentarz tekstowy

commenttime - czas wyświetlania komentarza

ctrl - kontroler

default - wartość wstępna

defaultposition - pozycja wstępna

defaultpositionx - pozycja wstępna dla x

defaultpositiony - pozycja wstępna dla y

elements - ilość elementów zawartych w bitmapie pokręta (knoba), każdy z takich elementów to zwykle przedstawia osobną pozycję pokręta

encoder - kółko „knob” pracujące jako dial (encoder)

fader - element suwak

fader: - obiekt suwak

fixedposition - opcja w definicji elementu powodująca, że dany element suwak, button, przycisk będzie miał zawsze stałą wartość w zakresie od 0.0 do 1.0

fixedwidth - umożliwia ustawienie stałej szerokości opisu text lub tekstu zmiennego **vtext**

font - definicja czcionki

grid - definiuje siatkę miksera (domyślnie jest to 1 x 1 piksel)

group - służy do grupowania obiektów, na przykład przycisków

hidemouse - ukrywa wskaźnik myszy na obiekcie

hoverbitmap - podświetlanie dla obiektów suwakowych „fader” i przycisków „button”

inverse - odwraca zakres pracy elementu

knob - element gałka

knob: - obiekt gałka

korg - rozbija zmienną w SySex na 14-bitowy format Korga

list - element lista

list: - obiekt lista

move - zakres ruchu suwaka, **move horiz** [zakres w pikselach]. Np. move horiz 100 oznacza że, ruch suwaka czyli bitmapy zmiennej

vbitmap po ekranie będzie odbywał się w zakresie 100 pikseli od lewej do prawej

nibble - opcja dzieląca bajt na tzw. pół-bajty
off - podcina komunikat SysEx lub CTRL pod stan wyłączenia przycisku dwustanowego
offset ofs - przesuwa grupowo obiekty na ekranie
on - podcina komunikat SysEx lub CTRL pod stan włączenia przycisku dwustanowego
onestate - wyłączenie trybu wciśnięcia dla przycisku button
page - definicja numeru i nazwy Strony
plane - definiuje tzw. przycisk planszowy, w zasadzie ten sam efekt uzyskamy zestawiając kilka przycisków obok siebie, opcja ta stworzona została dla oszczędności i przejrzystości kodu oraz wtedy gdy nie potrzebujemy większych kombinacji jak jeden SYSEX lub jeden CTRL pod jeden ELEMENT.
refreshbackground - opcjonalnie odświeża tło przycisku dwustanowego „button”
rolandcrc - funkcja licząca sumę kontrolną **crc** dla urządzeń firmy Roland
rows - dzieli rozwijalną listę na kolumny
scroll - obiekt, umożliwia wstawienie suwaków do skrolowania zawartości okna
selectlist - lista rozwijalna o skończonej ilości elementów. Jedna pozycja steruje tylko jednym komunikatem sysex lub ctrl
step - wartość krokowa
stretch - zmienia rozmiary bitmapy, powiększając lub zmniejszając ją o zadaną wartość
sysex - format komunikatów MIDI
table - dodaje listę opisową dla zmieniającej się zmiennej
table2 - dodaje listę opisową dla zmieniającej się drugiej zmiennej
text - tekst
texture - tło dla Strony
titlebar - ustawia własny tytuł okna programu
transparentclick - aktywuje klik na transparentnym tle obiektu
twobyte - rozбивa zmienną w SysEx na dwa bajty (LSB i MSB)
vbitmap - grafika dla poruszającego się elementu
vtext - wyświetlane dane, zmieniające się podczas regulacji
window - wymiary okna Strony
workarea - obszar roboczy
xaxis - podcina komunikat SysEx lub CTRL pod oś X elementu xycontrol
yaxis - podcina komunikat SysEx lub CTRL pod oś Y elementu xycontrol
xycontrol - element X-Y
xycontrol: - obiekt X-Y

Strona programu: www.extranslator.com/tools/exmixer.html

Inne strony autora programu: www.extranslator.com, www.samplelord.com

Kontakt: info@extranslator.com