

MIDI Synthesizer

Udo matthias 07626-2 999 847

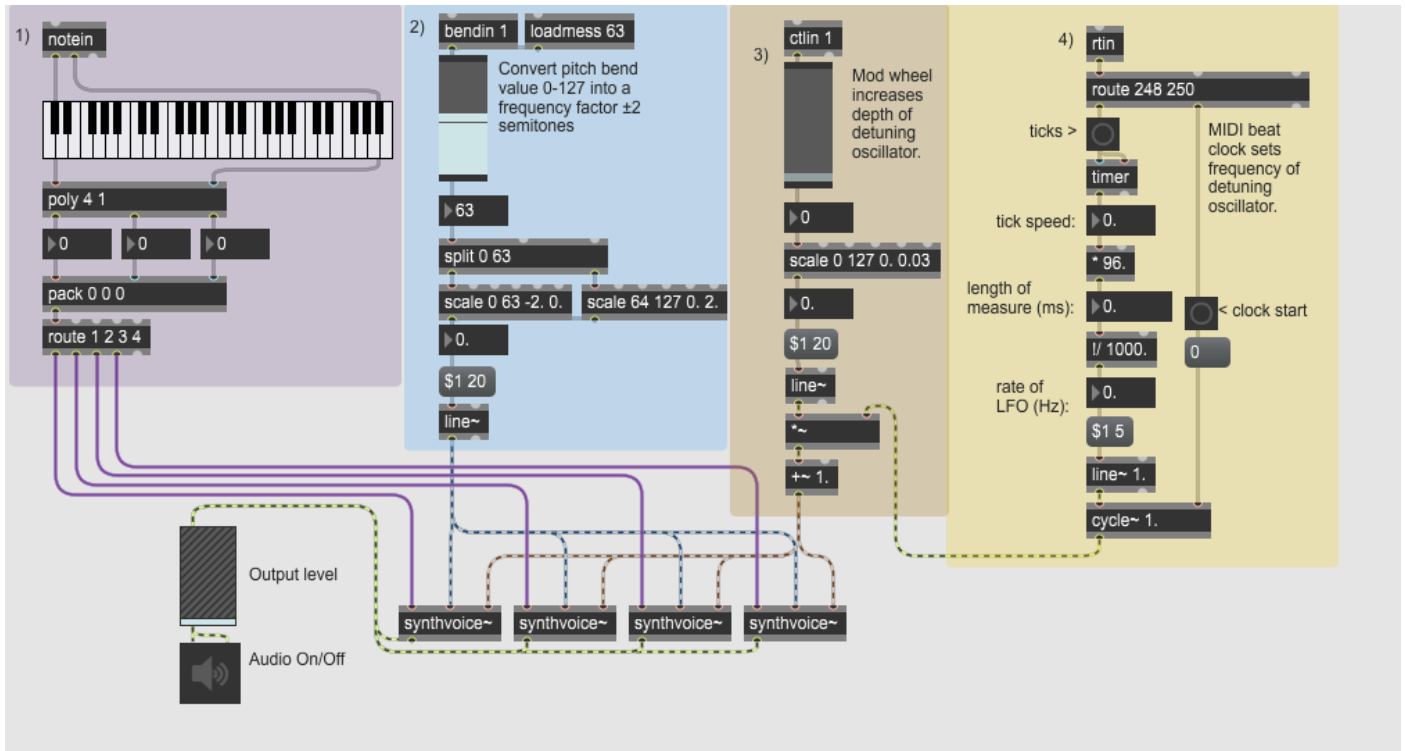
mobil: 017621-605276

<http://www.udomatthias.com>

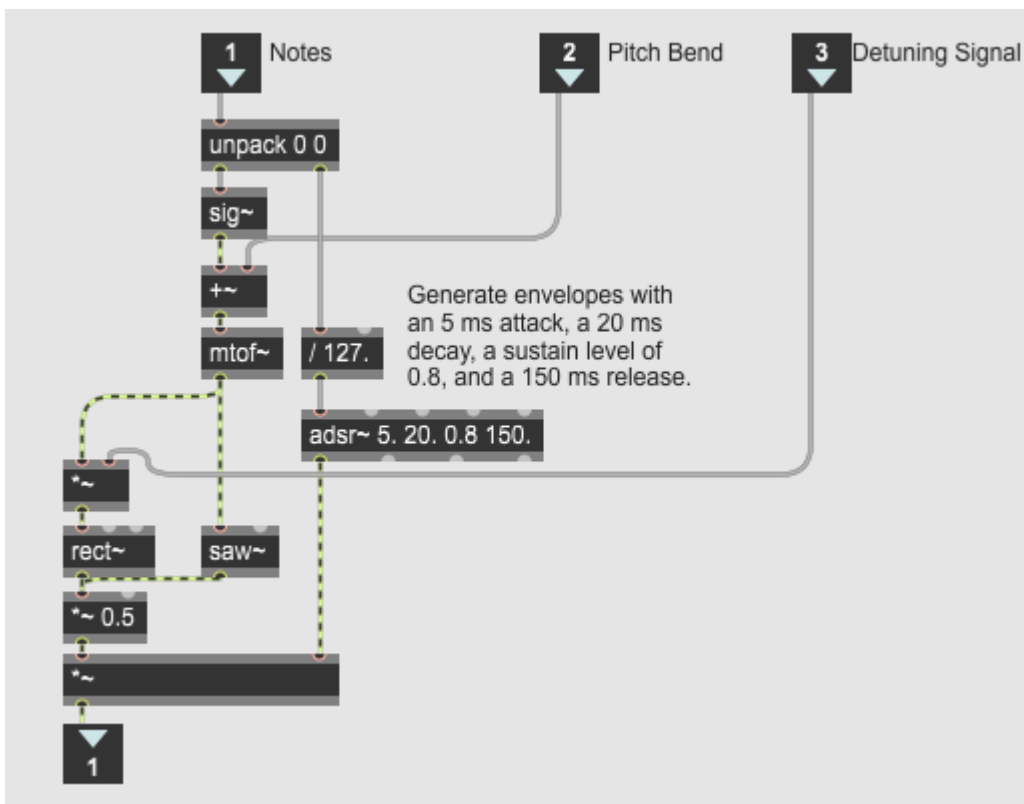
<https://www.facebook.com/udo.matthias.3>

info@udomatthias.com

If it feels good, it must be in time!!



Subpatch synthvoice~



MIDI Synthesizer

Udo matthias 07626-2 999 847

mobil: 017621-605276

<http://www.udomatthias.com>

<https://www.facebook.com/udo.matthias.3>

info@udomatthias.com

MIDI Tutorial 2: MIDI Synthesizer

If it feels good, it must be in time!!



In diesem Tutorial zeigen wir, wie man einen einfachen **4-stimmigen MIDI-steuerbaren Synthesizer** erstellt und damit arbeitet. Auf dem Weg dorthin diskutieren wir **einfache Methoden des polyphonen Routings** sowie die Verwendung einer **Vielzahl gängiger MIDI-Nachrichten zur Steuerung der Syntheseparameter**.

➔ Um das Tutorial-Patch zu verwenden, stellen wir sicher, dass ein korrekt konfigurierter MIDI-Controller an Ihren Computer angeschlossen ist. In den Tutorials in diesem Abschnitt werden verschiedene MIDI-Meldungen als Beispieleingabe verwendet. Wenn unserem Controller eine dieser Funktionen fehlt, können wir deren Eingabe mit Benutzeroberflächenobjekten im Lernprogramm simulieren.

Wir spielen unseren Synthesizer

➔Schauen wir uns zuerst den Patcher an. Beachte, dass es einiges gibt in der Patcher-Logik, die **vier verschiedene Kategorien von MIDI-Eingangsobjekten umfasst**, die Nachrichten und Signale an vier Kopien einer Abstraktion namens synthvoice~ senden.

➔Schalten wir also Audio ein, indem wir auf den ezdac~ klicken, den gain~ -Regler aufdrehen und einige Noten auf unserem MIDI-Keyboard spielen. Das kslider-Objekt sollte als Reaktion auf Ihre Aktionen animiert werden.

Das Pitch Bend-Rad kann man anpassen, während Sie eine Note halten. Wir stellen das Modulationsrad oder einen anderen Controller ein, der CC#1 senden kann. **Hören wir uns danach die verschiedenen Klangänderungen an.**

Unser Tutorial-Patcher behandelt MIDI auf verschiedene Arten.

➔Schauen wir uns diese nacheinander an.

➔Spielen wir nacheinander Noten auf unserer Tastatur und sehen uns den Bereich des **Tutorial-Patchers mit der Bezeichnung 1 an**. Beachte:, was **das poly-Objekt mit den Werten macht**.

Versuche, eine Note zu spielen und zu halten, und füge dann eine weitere hinzu, während die erste Taste noch gedrückt **ist**. ➔**Akkorde!!**

Unser Patcher kann vier Sounds gleichzeitig abspielen, da unser Patch **vier verschiedene Kopien der Synthvoice-Abstraktion enthält**. Um die **Polyphonie nutzen zu können**, müssen wir jedoch herausfinden, wie wir unsere MIDI-Werte an die verschiedenen Stimmen weiterleiten, damit die entsprechende Kopie der synthvoice~ -Abstraktion jede Nachricht empfängt.

Das Poly-Objekt nimmt die MIDI-Tonhöhe und -Geschwindigkeit an und führt eine Sprachzuweisung für die Werte basierend auf den Argumenten für das Objekt durch.

Die erste Note, die das Objekt empfängt, erhält die Stimmen Nummer 1, die zweite Note Stimme 2 und so weiter. Sobald wir die Polyphonie überschreiten, für die das Objekt programmiert

MIDI Synthesizer

Udo matthias 07626-2 999 847

mobil: 017621-605276

<http://www.udomatthias.com>

<https://www.facebook.com/udo.matthias.3>

info@udomatthias.com

ist (in unserem Fall 4 Stimmen), rollt das Objekt herum und beginnt erneut bei Stimme 1. Note-Off-Ereignisse werden der gleichen Stimme wie die entsprechenden Note-On-Ereignisse zugeordnet, was dies garantiert dass die Nachricht zum Stoppen eines MIDI-Events an dasselbe Ziel gesendet wird wie das, mit dem es gestartet wurde.

If it feels good, it must be in time!!



➔ **Spielen wir zuerst mit unserem MIDI-Keyboard einen Akkord mit vier Noten.** Dann drücken wir eine fünfte Note, ohne eine der Tasten loszulassen. **Beachte, was passiert.**

Das zweite Argument zu poly(1) besagt, dass Stimmen gestohlen werden sollen, wenn die Polyphonie überschritten wird. Wenn wir versuchen, mehr Noten zum Klingen zu bringen, als das Poly-Objekt zulässt, wird die letzte Note gelöscht und ihre Stimme wird recycelt.

Die Ausgabe für unser Polyobjekt wird in ein pack-objekt und dann über ein route-objekt gesendet, um die verbleibenden Listen von Tonhöhen und Geschwindigkeiten an die entsprechende Kopie unserer synthvoice-Abstraktion zu senden.

➔ Sehen wir uns den Tutorial-Bereich 2 an. Wir bewegen das Pitch bend-Rad an unserem Tastatur-Controller. **Beachte: dass sich der Ausgangspunkt im Gegensatz zu den meisten Controllern bei 63 in der Mitte des Bereichs befindet.** Wir sehen, wie die Patcher-Logik unten die Werte zwischen -2 skaliert. und 2.

Um unseren Pitch-bend-Wert zu erstellen, nehmen wir das MIDI Pitch bend-Rad und teilen seinen Bereich auf zwei verschiedene Skalenobjekte auf. Das linke Skalierungsobjekt nimmt die untere Hälfte des MIDI-Bereichs und skaliert sie von -2. bis 0.;

Das rechte Objekt skaliert den oberen Bereich zwischen 0 und 2. Dies garantiert, dass der Mittelwert im Bereich (63) in allen Fällen einem Wert von 0 entspricht.

➔ **Wenden wir uns dem Tutorial-Bereich mit der Bezeichnung 3 zu.** Wir bewegen den kontinuierlichen Controller und sehen, wie er skaliert ist.

➔ **Nun schauen wir den Tutorial-Bereich mit der Bezeichnung 4 an.** Wenn wir einen **MIDI-Controller mit Echtzeit-Transportfunktionen haben (d.h. MIDI-Beat-Clock senden können)**, stellen wir ihn so ein, dass er in einem beliebigen Tempo sendet, und starten dann den Transport auf dem Controller. Wenn wir möchten, können wir dazu auch ein **anwendungsübergreifendes MIDI-Routing-Dienstprogramm und einen Software-Sequenzier auf Ihrem Computer verwenden.** Wenn nicht, klicken wir in das Zahlenfeld mit der **Bezeichnung Tick Speed** und **geben Sie den Wert 20 ein.** Ändern wir nun den Regler und spielen einige Noten und hören uns das Ergebnis an. Ändern wir die Tick-Geschwindigkeit auf etwas schnelleres (wie 10.). Was passiert?

Die MIDI CC# - und die Echtzeitnachrichten werden von den Objekten ctlin bzw. rtin gesendet, um gemeinsam einen Niederfrequenzoszillator (LFO) zu steuern. Die Echtzeitnachrichten, die den Takt

MIDI Synthesizer

Udo matthias 07626-2 999 847

mobil: 017621-605276

<http://www.udomatthias.com>

<https://www.facebook.com/udo.matthias.3>

info@udomatthias.com

definieren, steuern die Frequenz (rate) des LFO. Die Controller-Meldungen ändern die Tiefe. Der MIDI-Beat-Takt läuft normalerweise mit 96bpm. Das Timer-Objekt misst die Intervalle zwischen Ticks, die dann skaliert werden, um die Frequenz des LFO so abzuleiten, dass sie einen Takt dauert. **Die MIDI-Echtzeitnachricht 250 (die 'Start'-Nachricht auf einem Sequenzer) setzt die Phase des Zyklusobjekts zurück, so dass der LFO erneut mit einer Sequenz synchronisiert wird, wenn er auf einer Taktzeile startet.**

If it feels good, it must be in time!!



Beachte: Wie wird die Ausgabe des LFO so skaliert, und dass selbst bei einer Tiefe, die ein Signal von 0 sendet, das in die synthvoice~ - Abstraktion gesendete Signal garantiert dass 1 zentriert ist.

Schauen wir uns kurz an, was in diesem Patch enthalten ist.

Die synthvoice~ Abstraktion (Subpatch (UP))

→ Doppelklicken wir auf eine der Abstraktionen mit dem Namen synthvoice~ und sehen uns die patcher-Logik an.

Unsere synthvoice~ Abstraktion hat **drei inLets s. später -Javascript).**

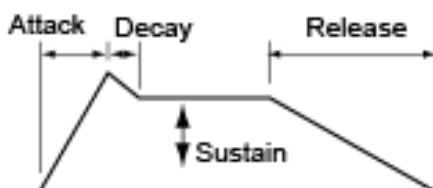
Der erste Eingang nimmt Listen mit Tonhöhen- und Velocity-Werten von unserem MIDI-Keyboard-Eingang. Der Tonhöhenwert wird gesendet, um ein konstantes Signal (sig~) anzusteuern, dem das Signal vom Einlass Nr. 2 hinzugefügt wird (+~)Addition. Dieser Wert wird dann als MIDI-Nummer interpretiert und von einem Objekt namens mtof~ in eine Frequenz im Signalbereich umgewandelt, das sich wie das mtof-Objekt verhält, jedoch mit kontinuierlichen MSP-Signalen anstelle von Max-Nummern arbeitet.

Dieser Frequenzwert speist dann zwei bandbegrenzte Oszillatoren:

- eine Rechteckwelle (Rechteck) und eine Sägezahnwelle (Säge), die miteinander gemischt werden. Die Frequenz des Rechteckobjekts wird mit dem in Eingang Nr. 3 eingehenden LFO-Signal multipliziert, so dass ein satter, Ton erreicht werden kann, wenn die Tiefe des LFO erhöht wird.

Standard-Synthesizer-Hüllkurven: adsr~

Währenddessen wird der Velocity-Ausgang der MIDI-Noten zwischen 0 und 1 skaliert und an ein Objekt namens adsr~ gesendet. Das Objekt steht für Attack, Decay, Sustain, Release und generiert Signalarmpen in einer Standardkonfiguration, die aus dem analogen Synthesizer-Design entlehnt wurde:



A standard ADSR curve.

Die Argumente für adsr ~ werden als Angriffszeit,

MIDI Synthesizer

Udo matthias 07626-2 999 847

mobil: 017621-605276

<http://www.udomatthias.com>

<https://www.facebook.com/udo.matthias.3>

info@udomatthias.com

Abklingzeit, Sustain-Level und Release-Zeit interpretiert. Der Sustain-Pegel ist ein Multiplikator der Gesamtamplitude, die das Objekt während einer Sustain-Note ausgibt.

If it feels good, it must be in time!!



Das `adsr~` -Objekt nimmt einen Wert an und interpretiert ihn als Hüllkurventrigger einer bestimmten Amplitude. Jede Zahl höher als 0 löst die Attack-, Decay- und Sustain-Phasen aus, die so skaliert sind, dass sie mit der Amplitude des Triggers übereinstimmen (z. B. löst ein Eingabewert von 0,5 eine weichere Hüllkurve aus als ein Trigger von 0,8). Das Objekt bleibt dann in der Sustain-Phase und gibt einen konstanten Wert aus, bis es eine 0 erhält. Es fährt dann mit der Freigabephase der Hüllkurve fort und steigt auf 0 an.

➔ Nachdem wir nun verstanden haben, wie die Synthesizer-Elemente funktionieren, kehren wir zum Haupt-Patch zurück und prüfen, ob die steuerbaren Parameter sinnvoll sind. Erkunden wir also den gesamten Bereich des MIDI-Reglers, um zu sehen, wie ausdrucksstark es ist.

Zusammenfassung

Innerhalb eines Patches können MIDI-Notenereignisse mithilfe eines Polyobjekts `polyphon` an verschiedene Kopien derselben Abstraktion weitergeleitet werden.

Objekte wie `bendin` und `ctlin` können skaliert werden, um verschiedenen Synthesizer-Parameterbereichen zu entsprechen, und **Echtzeit-MIDI-Befehle** können verwendet werden, um Tempodaten für LFOs und Sequenzer innerhalb von MAX zu generieren. Das `adsr~` -Objekt generiert Hüllkurvenrampen basierend auf Triggern für ein Note-On und ein Note-Off, was es ideal für die Verwendung mit MIDI-Note-basierten Systemen macht.

Siehe auch

Name	Description
poly	Allocate notes to different voices
mtof~	Convert a MIDI note number to frequency at signal rate
adsr~	ADSR envelope generator

[MSP MIDI Tutorial 1: Mapping MIDI to MSP](#)

[MSP Tutorials: Table of Contents](#)

[MSP MIDI Tutorial 3: MIDI Sampler](#)

Echtzeit: