

SensOrchestra

Sensor-zu-MIDI-Konverter

Bedienungsanleitung

Stand: Nov.2004

Einführung

SensOrchestra (in der Folge als SO bezeichnet) ist ein Sensor-zu-MIDI-Konverter auf Mikrocontrollerbasis. Es ist konzipiert zur Integration sowohl in existierende Instrumente, als auch in eigene Entwürfe. Deshalb wird es zwar betriebsfertig, jedoch ohne Gehäuse geliefert. Haupteigenschaften sind die kleine Baugröße, Sensibilität in musikalischem Sinne, gegeben durch hohe Abtastgenauigkeit und Schnelligkeit (geringe Latenz) sowie einfache Erweiterbarkeit. Die Grundkonfiguration bietet 16 analoge Eingänge, wobei diese Anzahl durch Hinzufügen weiterer SO-Module sehr einfach nahezu beliebig erhöht werden kann. Eingangsseitig können einerseits widerstandsveränderliche Signalquellen, wie druck- oder lichtempfindliche Sensoren (FSR, LDR) oder Potentiometer, andererseits Spannungen (CV) verwendet werden. Am Ausgang können nahezu alle definierten Midi-Nachrichten erzeugt werden. Der Konvertierungsprozeß wird durch eine Vielzahl von Parametern gesteuert, welche benutzerseits mittels Midi System-exclusive-Nachrichten (Sysex) einstellbar sind. Der Einsatz von SO erfordert keine speziellen technischen Fähigkeiten. Für Änderungen an der Konfiguration sind jedoch Kenntnisse von Midi Sysex und entsprechender Software, sowie der hexadezimalen Zahlendarstellung erforderlich.

Benutzungshinweise

Vorsichtige Behandlung zur Vermeidung mechanischer Schäden beim Einbau wird ebenso vorausgesetzt wie Schutz vor elektrischer Beschädigung durch Kurzschluß, Überspannung oder ungeeignete Eingangssignale. Insbesondere dürfen Piezo-Sensoren NICHT verwendet werden, da sie die Schaltung zerstören können. Die Module sollten beim Einbau an geschützter Stelle so positioniert werden, daß die LEDs sichtbar bleiben, und dort gut befestigt (angeklebt, angeschraubt) werden. Auch die Zugentlastung der Stecker sollte gewährleistet sein. Die Kabel zwischen den SO-Komponenten sowie zu den Sensoren sollten so kurz wie möglich bemessen werden; Maximallängen sind schwer anzugeben, da die elektrischen Umgebungsbedingungen wie auch die Anzahl der SO-Module eine Rolle spielen können. Sensorkabel bis 1 Meter scheinen zwar verlässlich, im Zweifelsfall sind jedoch entsprechende Tests ratsam. Die Sensoren sollten einzeln Stück für Stück angeschlossen und getestet werden, da eventuelle Fehler so leichter zu entdecken sind. Besonders wichtig ist der sorgfältige Umgang mit den gut geeigneten aber auch empfindlichen FSR-Sensoren (Hersteller IEE). Sie sollten auf möglichst ebener, nicht gebogener Fläche befestigt werden. Keinesfalls sollten sie geknickt oder mit harten Gegenständen gespielt werden!

Teilebeschreibung

Ein SO-Basissystem besteht aus aus folgenden Elementen:

1. SObar, die zentrale SO-Komponente trägt zwei Mikrocontroller(SOchips) mit je 8 analogen Sensor-Eingängen an den Steckern A und B. Aus den dort gemessenen Sensor-Eingangswerten werden Mididaten erzeugt. (Informationen über Systeme mit mehr als einem SObar siehe unter "Erweiterung" am Ende der Anleitung)
2. Midi-power-Adapter enthält Anschlüsse für Midi-Ausgang und Midi-Eingang (für Konfigurationszwecke) sowie für die Stromversorgung. Der Midi-Ausgang dient gleichzeitig als Midi-Thru.
3. Flachbandkabel 10-polig, zur Verbindung dieser beiden Teile.

Für ein komplett funktionsfähiges System werden weiters benötigt:

4. Steckernetzteil, stabilisiert, Stecker 2.1mm, DC 7,5V, 300mA (für ein SObar; bei Anschluß mehrerer SObars stärkeres Netzteil verwenden)
5. Sensorkabel; üblicherweise 16-polige Flachbandkabel, evtl. auch Einzelkabel, je nach Einbauentfernung der Sensoren
6. Sensoren: widerstandsändernde Typen wie FSR, LDR, Potentiometer etc.; (auch Spannung 0-2.5 V möglich)
7. Midi-Kabel
8. Midi-Empfänger: Synthesizer, Sampler, Computer

Die Teile 4, 5, 6, 7 werden je nach Bedarf bzw. Vereinbarung mitgeliefert oder müssen benutzerseits beschafft werden.

Installation und Funktionstest

[SObar und Midi-power-Adapter sollten im Lieferzustand verbunden sein. Andernfalls mittels 10-poligen Flachbandkabels verbinden.]

1. Sensor an SObar anschließen: dazu 16-poliges Sensorkabel in SObar-Anschluß A oder B und Sensor an andere Kabelende stecken (zum Testen vorerst nur einen Sensor). Falls 2-poliges Kabel für Einzelsensor verwendet wird, eines der Pin-Paare 1 - 8 lt. Beschriftung benutzen.
2. Midi-out des Midi-power-Adapters mit Midi-in von Tonerzeuger oder Computer mittels Midi-Kabels verbinden.
3. Stromversorgung: (vorher Lautstärke zurückdrehen!): Steckernetzteil mit Midi-power-Adapter verbinden, dann Netzteil einstecken.

Jetzt sollten am SObar zwei rote LEDs für ca. 2 Sekunden aufleuchten, anschließend dann die gelbe und grüne LED permanent leuchten. Sollte dies nicht geschehen, bitte Strom ausstecken, Polarität der Stromversorgung und alle Verbindungen prüfen. Bleibt das Problem bestehen, bitte rückfragen!

Falls jedoch alles wie erwartet funktioniert, sollte bei Betätigung des Sensors der Midi-Klangerzeuger zu hören oder Midi-Daten am Midi-Monitor zu sehen sein; in der Standardkonfiguration sendet SO Noten auf Midi-Kanal 1. Die Helligkeit der roten LED entspricht dabei der Stärke des Sensorsignals. Wenn alle Sensoreingänge fehlerfrei arbeiten, sollte SO bereit zum Einbau und zur produktiven Verwendung sein.

Standardeinstellung

SO enthält vier vorkonfigurierte Programme (SOprg, sogenannte "SO-Progs"), welche folgende Midi-Daten erzeugen:

- SOprg 1: note on/off, chip A: Notenummer 36-43, chip B: 44-51 (d.h. ab Note c0 chromatisch steigend)
- SOprg 2: continuous controller, chip A controller Nummer 1-8, chip B controller Nummer 17-24
- SOprg 3: note on/off und CC, d.h. SOprg 1 und 2 zugleich
- SOprg 4: note on/off, aber Noten umgekehrt wie bei SOprg 1

Im Lieferzustand wird beim Einschalten SOprg 1 gewählt, womit Noten auf Midi-Kanal 1 erzeugt werden. Zwischen den SOprgs kann durch Midi Program Change Nachrichten (program 1 - 4) auf Midi-Kanal 1 umgeschaltet werden.

Die Standardkonfiguration mag für den Anfang reichen, oder auch bei Verwendung von SO als Midi-Signalquelle für Pakete wie PD oder Max/MSP, in welchen Midi-Daten modifiziert werden können. Üblicherweise wird freilich der Wunsch bestehen, SO den eigenen Bedürfnissen anzupassen. Es folgt ein Überblick über die derzeit dafür gegebenen Möglichkeiten.

Parameter und Software-Konfiguration

[Hexadezimale (hex) Zahlen werden in der Folge wie üblich durch vorangestelltes "0x" gekennzeichnet.]

Die Funktionen von SO sind benutzerseitig durch Parameter in weitem Bereich einstellbar. Dies umfaßt die Analyse der Sensor-Eingangswerte, die Erzeugung der Midi-Daten sowie diverse weitere Funktionen. Beispiele für solche Parameter sind Midi Noten-Nummern, Midi controller-Nummern, Midikanäle, Rauschpegel, Entprellung, Kurven für die Zuordnung von Eingangswerten zu Midi-Ausgangswerten, Midi-Basiskanäle etc. Es gibt über 30 Parametertypen und die Gesamtzahl der einstellbaren Einzelparameterwerte pro SOchip liegt bei ca.550. Die Parameteränderungen erfolgen durch Midi System exclusive (Sysex) Nachrichten.

Da das Verständnis des SO-Konzepts für den Konfigurations- oder Editiervorgang unerlässlich ist, folgt ein kurzer Überblick. Die kleinste SO-Funktionseinheit ist ein SOchip mit den 8 dazugehörigen Sensoren bzw. Eingängen. Die gegenwärtige Bauform SObar mit zwei SOchips wurde lediglich aus praktischen Erwägungen gewählt und ist für den Konfigurationsprozeß ohne Belang. Somit läßt sich das SO-Konzept folgendermaßen als Drei-Ebenen-Modell darstellen:

1. Einzelner Sensor/Eingang
2. SOchip
3. SO-Instrument (ein Midi-power-Adapter mit einer Kette beliebig vieler SObars bzw. SOchips)

Im Spiel- oder Performance-Modus bleibt die Ebene 2 (SOchip) außer Betracht; als BenutzerIn interagiert man nur mit den Ebenen 1 und 3 (Sensor und SO-Instrument). Um aber im Editiermodus einen Parameterwert eines bestimmten Sensors an einem bestimmten SOchip ändern zu können, muß dieser eindeutig adressierbar sein. Eine vollständige Adresse besteht aus:

(a) der SOchip-Identitätsnummer SOchip-Id

- (b) der Nummer des Sensor-Eingangs an diesem SOchip und
- (c) der Parameter-Nummer.

Wie kommt man nun zu diesen Adressinformationen?

zu (a) Jeder SOchip muß zu Beginn des Editiervorganges durch eine Sysex-Nachricht eine eindeutige SOChip-Id zwischen 1 und 123 (0x01-0x7B) erhalten, sinnvollerweise von 1 weg aufsteigend. Damit wird der unterschiedlichen Anzahl von SOchips in verschiedenen SO-Instrumenten Rechnung getragen. (Siehe auch unter "Erweiterung") Beim Einschalten tragen alle SOchips die SOchip-Id 0.

zu (b) Die Sensornummer entspricht dem entsprechenden Eingang (0x00-0x07) am jeweiligen SOchip.

zu (c) Die gewünschte Parameter-Nummer ist Tabelle 1 im Anhang zu entnehmen.

Zum Editieren müssen Midi-Verbindungen vom Editor-PC zum SO-Midi-Eingang sowie vom SO-Midi-Ausgang zum Editor-PC hergestellt werden.

Mit den Adressdaten lassen sich dann Sysex-Nachrichten der Art "Ändere Parameter 0x23 (=Notennummer) bei Sensor 5 an SOchip 2 auf den Wert 0x54" an SO schicken. SO antwortet darauf mit einer Sysex-Nachricht, welche einen Antwortcode zur Signalisierung der korrekten Verarbeitung oder etwaiger Fehler (ungültiger Wert oder Parameter) enthält. Die SOchips sind in Serie geschaltet; jeder SOchip erkennt anhand der SOchip-Id Nachrichten, die für ihn bestimmt sind und verarbeitet sie; die übrigen werden unverändert durchgeschleift.

Alle Parameterwerte lassen sich darüber hinaus auch ohne vorherige Änderung einfach abfragen (damit ist kompletter dump request möglich).

Editieren der Parameter

Der Editiervorgang umfaßt das Einlesen, Darstellen und Verändern der Parameterwerte im SO. Weiters wäre das Laden von und das Speichern auf externem Speicher (Festplatte) nützlich. Ein graphischer Editor für SO wäre dafür die Ideallösung, existiert jedoch gegenwärtig nicht. (Ein entsprechendes Projekt in Java wurde zwar vor einiger Zeit begonnen, die Fertigstellung ist aber im Moment nicht absehbar.) Sollte jemand interessiert daran sein, einen Editor zu schreiben, so steht die SO-Spezifikation sowie Unterstützung gerne zur Verfügung!

Somit ist man zur Zeit also darauf angewiesen, die entsprechenden Sysex-Nachrichten manuell zu erstellen und mittels eines geeigneten Programms (Midiox, Cubase,...) an SO zu schicken. Die Antwort von SO sollte anschließend geprüft werden. Man kann auch Sysex-Dateien mit mehreren Nachrichten erstellen, speichern und abschicken. Dabei sollte beim Versenden eine Pause von ca.20 Millisekunden zwischen den einzelnen Nachrichten eingehalten werden.

Als Zwischenlösung kann das Tcl/Tk-Programm SoTools (für Windows) zur Verfügung gestellt werden. Dieses bietet gewisse Unterstützung beim Editieren (automatische Vergabe der SOchip-Id, Prüfung der Antwort-Meldungen, Fehlerausgabe, dump request und Speichern der Parameterwerte, Kopieren der Gesamtkonfiguration auf anderes SO-Instrument.) Bei Bedarf bitte nachfragen!

Weiters können nach Vereinbarung auch fertige Konfigurationsdateien zur Verfügung gestellt werden.

SensOrchestra Sysex-Format

Diese Zusammenfassung enthält jene Informationen über das SO-Sysex-Format, welche zum Editieren notwendig sind.

Eine SO-Sysex-Nachricht besteht aus 17 Bytes folgender Bedeutung (alle Werte hexadezimal):

Bytes 1-6: F0 7D 46 48 01 01 - konstant (Start Sysex, Herstellerkennung, Version)

Byte 7: report class, kommt in SO-Antwort: 10=okay, sonst Fehler

Bytes 8: report number, kommt in SO-Antwort; details zu report class

Bytes 9: target device Id: für wen ist Nachricht bestimmt?

Bytes 10: source device Id: wer hat Nachricht abgeschickt?

(Nummernbereiche: 01-7B): SOchip, 7C-7E:master (PC), 7F:broadcast=an alle)

Byte 11: instruction, Werte: 61=adjust & request (an SO: Wert ändern, lesen und als Antwort schicken), 21=request (an SO: Wert nur lesen und als Antwort schicken), 11=reply (von SO: Nachricht ist die Antwort von SO)

Byte 12: Parameternummer; siehe Tab. 1

Bytes 13-14: Index und Sub-Index zu Parameter, wo nötig (z.B. SOPrg-Nr. und Sensornummer); siehe Tab. 1

Bytes 15-16: Parameterwert, geteilt in nibbles, um Werte größer als 7F (Limit für Midi-Datenbytes) senden zu können; siehe Tab. 1

Byte 17: Ende Sysex

Ein praktisches Beispiel mit drei Sysex-Nachrichten könnte so aussehen:

```
F0 7D 46 48 01 01 00 00 00 7C 61 40 00 00 00 01 F7
```

```
F0 7D 46 48 01 01 00 00 00 7C 61 40 00 00 00 02 F7
```

```
F0 7D 46 48 01 01 00 00 01 7C 61 23 03 02 04 05 F7
```

Die ersten beiden Nachrichten setzen zwei SOchips vom Einschaltzustand 0 auf SOchip-Id 1 und 2. Dieser

Nachrichtentyp muß nach dem Einschalten nur ein mal geschickt werden.

Die dritte message setzt die zu sendende Notennummer bei SOchip 01, Programm 03, Sensor 02 auf 45.

Bei fehlerfreier Verarbeitung kommen folgende Nachrichten zurück:

F0 7D 46 48 01 01 10 00 7C 01 11 40 00 00 00 01 F7

F0 7D 46 48 01 01 10 00 7C 02 11 40 00 00 00 02 F7

F0 7D 46 48 01 01 10 00 7C 01 11 23 03 02 04 05 F7

Bytes 7 und 8 (10 00) signalisieren die erfolgreiche Verarbeitung; Bytes 9 und 10 sind jetzt vertauscht, da die SOchips als source (Quelle) Nachrichten an target 7C (master, PC) senden; laut Byte 11 sind diese vom Typ reply (Antwort). Bytes 12-16 enthalten Parameter, Index, Subindex und gelesenen Wert (nach Änderung).

SensOrchestra-Editierparameter

Tabelle 1 enthält eine Liste derjenigen SO-Parameter, die für den Editiervorgang von Bedeutung sind. Diese fallen je nach ihrer Funktionen in eine der folgenden Kategorien:

- a) allgemeine Midi-Konfiguration: Parameter 01, 03, 04, 05
- b) Midi-Ausgabe-Steuerung: Parameter 20-27
- c) Meßwerterfassung und Konvertierung: Parameter 02, 06-0F
- d) sonstige / temporäre Parameter: Parameter 40-42, 60

Parameterwerte der Kategorien a), b) und c) lassen sich dauerhaft in SO speichern, während die Parameter unter d) ihren Wert mit dem Ausschalten verlieren.

Parameteränderungen für Parameter 01, 03, 06, 07 erfordern Aus- und Einschalten, um wirksam zu werden.

Erläuterungen zu Parameter 02: Der Parameter dient dazu, die Empfindlichkeit der Sensoren der eigenen Spielweise anzupassen. Dazu können die Sensormeßdaten, dargestellt als Werte im Bereich 00-FF mittels einer Kurve aus 17 Punkten und Interpolation auf Midi-Werte von 0-127 abgebildet werden. Die Zuordnung kann z.B. linear, logarithmisch, in fixen Stufen, im Prinzip aber beliebig sein. Es können z.B. auch zwei Sensoren von einer fixen Mittelposition (Pitchbend!) aus nach oben und nach unten gehen (Meßwerte 0-255 --> Zuordnung auf Midi-Werte 64-127 bzw. 63-0).

Für erste anschauliche Editier-Tests empfehlen sich die Parameter im Bereich 20-27 (Midi-Ausgabe-Steuerung), z.B. 23 (Notennummer) wie in obenstehendem Beispiel.

Erweiterung

Zur Erweiterung eines SO-Systems stehen zwei Wege offen. Zum einen kann man einfach zusätzliche komplette SO-Systeme in die Midi-Kette hängen. Dies hat den Vorteil, daß man mit normalen Midi-Kabeln auch weitere Distanzen (bis 15m) überbrücken kann (z.B. für Rauminstallation). Zum anderen ist SO jedoch so konzipiert, daß ein Basissystem (=Midi-power-Adapter und SObar) durch dazustecken weiterer SObars auf nahezu beliebig viele Eingänge erweitert werden kann (größtes System bisher: 24 SOchips, 192 Sensoren). Diese Lösung besitzt den Vorteil geringerer Latenz und Baugröße. Die SObars werden dazu über Flachbandkabel verbunden. Für das Editieren ist konzeptbedingt davon auszugehen, daß alle SOchips innerhalb einer Midi-Kette ungeachtet der Art ihrer Verbindung als ein SO-Instrument betrachtet werden. Bei größeren Systemen ist auf genügende Stromversorgung (stärkeres Netzteil), sowie bei kompaktem Einbau auf entsprechende Luftzirkulation zur Kühlung zu achten.

SensOrchestra Editierparameter

Parametername	Param. nummer	Index	Subindex	Werte-bereich	Erläuterungen
PwrUpSoPrgNr	01	00	00	00-03	SoPrgNr. nach Einschalten
MeasToMidiMapTables	02	00-05	00-10	00-FF	6 Kurven/Tabellen mit je 17 Werten (Meßwert -> MIDI)
PwrUpUseRunningStatus	03	00	00	00-01	Running status für Midi-Ausgabe von SO: 0=off, 1=on
MidiBasicChannel	04	00	00	00-0F	Midi Basiskanal
SoPrgAssignedToPrgChg	05	00-7F	00	00-03	Zuordnung Midi Prg.change (00-7F) auf SoPrgNummer (00-03)
SensorType	06	Sensor	00	00-FF	Sensortyp: 01=FSR; 00=kein Sensor
PwrUpAdcSamplingPeriod	07	00	00	00-07	Sample rate (Standardwert = 06 --> 2.1 kHz/Sensor)
IgnoranceLevel	09	Sensor	00	00-FF	unterer Schwellmeßwert für Sensormessung (Rauschpegel)
TrigInertia	0A	Sensor	00	00-FF	Trigger-Verzögerung (Trigger-Bedingung muß sovielen samples lang bestehen)
ReEnableLevel	0B	Sensor	00	00-FF	Schwellwert (=Parameterwert mal 4) muß vor neuerlichem Trigger unterschritten werden
ReEnableDelay	0C	Sensor	00	00-FF	Mindestverzögerung (in samples) vor neuerlichem Trigger
MinSlopeCausingTrig	0D	Sensor	00	00-FF	Wertänderung (zwischen 2 Folge-samples) kleiner als dieser Wert löst Trigger aus
MinContTimeDiffOfValGrain	0E	Sensor	00	00-7F	Mindestzeit (=Parameterwert mal 32 samples) zwischen continuous messages
MaxContValGrain	0F	Sensor	00	00-7F	Änderung stärker als dieser Wert (als Midi-Wert) löst cont.msg vor Mindestzeit aus
SingleMsgMode	20	SoPrgNr	Sensor	00-7F	00=single message aus, 01=NoteOn/NoteOff, 0F=PrgChange
SgEvntMidiChannel	21	SoPrgNr	Sensor	00-0F	Midikanal für single message
VelocityMapTableNr	22	SoPrgNr	Sensor	00-05	Empfindlichkeitskurve für single message lt. Parameter 02
MidiNotePrgChg	23	SoPrgNr	Sensor	00-7F	single message Nummer (Note oder Midi Prog.nr. je nach Param. 20)
ContinuousMapTableNr	24	SoPrgNr	Sensor	00-05	Empfindlichkeitskurve für continuous message lt. Parameter 02
ContMsgTypeAndNr	25	SoPrgNr	Sensor	00-82	00-7F=Controller, 80=PolyAftert.(Note v. Param.23), 81=ChannelAftert., 82=Pitchbend
ContinuousMsgMode	26	SoPrgNr	Sensor	00-7F	00=continuous message aus, 01=ein
ContinuousMidiChannel	27	SoPrgNr	Sensor	00-0F	Midikanal für continuous message
Deviceld	40	00	00	00-7B	SoChip-Id zur Adressierung, vor Editieren zu setzen (Startwert=0)
SoPrgNr	41	00	00	00-03	aktuelle SoPrgNr (Startwert aus Parameter 01 gelesen)
IdentifyNextActiveSensor	42	00	00	00-01	01=Sensoridentifikation ein, schickt komplette Adresse lt. Param.60
SensorActivity	60	SoPrgNr	Sensor	00	Sensor schickt komplette Adresse (SoChipId+SoPrgNr+Sensor) wenn Param.42 = 01

Anmerkungen:

- alle Werte hexadezimal
- in Spalten Index/Subindex: Wert "Sensor" äquivalent zu 00-07, Wert "SoPrgNr" äquivalent zu 00-03
- single messages / Typ und Statusbyte: NoteOn (9x), Note Off(8x), Program Change (Cx)
- continuous messages / Typ und Statusbyte: continuous controller(Bx), polyphoner Aftertouch(Ax), channel Aftertouch(Ax), Pitchbend(Ex)
- "Trigger" bedeutet Erzeugung einer Midi single message
- "sample" = einzelne Messung

Tabelle 1: Liste der SensOrchestra Editier-Parameter