

MIDI Sensorbox

Betriebsanleitung für ab Februar 2007 gelieferte Geräte

Status 01. Februar 2007

Kommunikation und Stromversorgung:



IN ist ein Standard-MIDI Eingang.

OUT ist ein Standard-MIDI Ausgang.

Mit dem Codierschalter "Channel" wird der MIDI -Kanal für Befehle und Zustandsmeldungen ausgewählt.

Darüber hinaus werden in den Stellungen D, E, F und 0 **spezielle Möglichkeiten der Kommandosprache** aktiviert.

In den Stellungen 1 bis D (dezimal 13) ist der Standard-Kommandointerpreter aktiv:

Der Standard-Interpreter verarbeitet beliebig gemischt sowohl **MIDI Kanalnachrichten** als auch **textorientierte System-Exclusive Nachrichten**. Rückmeldungen auf Kanalnachrichten erfolgen als Kanalnachrichten und Rückmeldungen auf SysEx Nachrichten kommen als SysEx Nachricht mit Textinhalt zurück. Mit einfachen Befehlen können alle Eingangssignale ausgelesen werden, automatische Zustandsmeldungen aktiviert und Ausgangskonfigurationen eingestellt werden.

In Codierschalter-Stellung "D" arbeitet die Sensorbox ausnahmsweise mit 19200 Baud und mit reinen ASCII-Textbefehlen, d.h. SysEx Header und MIDI-Kanalnachrichten werden nicht ausgeführt. **Der Anschluss der MIDI-Buchsen an eine Standard RS-232 Schnittstelle ist mit einem optional lieferbaren Adapter oder mit einem leicht selbst herstellbaren Adapterkabel möglich.**

Zusätzliche Möglichkeiten bietet der **programmierbare Kommandointerpreter**, der in den Stellungen "E", "F" und "0" eingeschaltet wird.

Diese alternative Betriebsart der Sensorbox eignet sich vor allem für Standalone-Anwendungen, sowie um aus Sensorsignalen komplexere MIDI-Steuersequenzen herzuleiten oder umgekehrt mit einfachen MIDI-Befehlen komplexere Sensor/Aktuator-Operationen auszulösen.

Eine kurze Beschreibung dieses Interpreters finden Sie in Anhang B, für die ausführliche Beschreibung gibt es ein separates Manual.

Zur Einbindung in verschiedene Programmierumgebungen arbeitet die Sensorbox in Codierschalter-Stellung "E" mit 19200 Baud, in Stellung "F" mit 115200 Baud und in Stellung "0" mit MIDI Datenrate.

Die **Leuchtdiode** zeigt das Vorhandensein der Betriebsspannung an. Wenn die Box eine Nachricht empfängt, blinkt die Lampe kurz dunkel.

Die **Stromversorgung** ist ausgelegt für einfache unstabilisierte Gleichspannungs-Netzteile mit mindestens 250 mA Belastbarkeit. Ein einfaches Steckernetzteil mit "Eurostecker" ist im Lieferumfang.

Insgesamt sind die Anforderungen an die Stromversorgung unkritisch. Die Speisung kann auch aus anderen stabilisierten und unstabilisierten Gleichspannungsquellen erfolgen (konzentrischer DC-Stecker, aussen 5,0 - 5,5mm, innen 2,1mm). Die Spannung am Netzteil sollte auf 9 oder 12 Volt eingestellt sein, Spannungen bis max. 16 Volt (unstabilisiertes Steckernetzteil eingestellt auf 12 Volt) sind akzeptabel.

Wenn Anschluss **K6 als digitale I/O Leitung** konfiguriert ist (Lieferzustand), empfehlen wir wegen der geringeren Erwärmung eine Speisespannung von 9 Volt. Wird dieser Anschluss als **Analogausgang** konfiguriert (Details siehe unten), sollte die Speisespannung 12 Volt betragen.

Der **Pluspol der Spannung muss am inneren Kontakt** des DC-Steckers liegen!

Intern verfügt die Sensorbox über einen **Verpolungsschutz**: bei falsch gepoltem Netzteil wird er nicht eingeschaltet.

Aktuator/Sensor-Anschlüsse an der Rückseite



Acht sechspolige Stecksockel für **steckbare Schraubklemmen** sind montiert, Rastermaß 3,5 mm.

Auf der linken Gruppe Stecksockel sind die 12 kombinierten Digital I/O herausgeführt. (Die entsprechenden Signalanschlüsse sind nachfolgend mit K0 bis KA bezeichnet.)

Es können sowohl 3-polige als auch 6-polige Steckerteile verwendet werden. Standardmäßig liefern wir hierfür 4 sechspolige Steckerteile mit Schraubklemmen.

Socket K0 | K1/2: Pin1 = Masse, Pin2 = K0, Pin3 = +5V,
Pin4 = Masse, Pin5 = K1, Pin6 = K2

Socket K3 | K4/5: Pin1 = Masse, Pin2 = K3, Pin3 = +5V,
Pin4 = Masse, Pin5 = K4, Pin6 = K5

Socket K6 | K7/8: Pin1 = Masse, Pin2 = K6, Pin3 = +5V,
Pin4 = Masse, Pin5 = K7, Pin6 = K8

Socket K9 | KA/B: Pin1 = Masse, Pin2 = K9, Pin3 = +5V,
Pin4 = Masse, Pin5 = KA, Pin6 = KB

Auf der rechten Gruppe Stecksockel sind die 8 analogen Spannungseingänge in Dreier-Kombinationen herausgeführt.

Der linke Kontakt jeder Dreier-Kombination ist Masse, der mittlere Kontakt ist der Spannungseingang, am rechten Kontakt ist die 5 Volt Spannung zur Versorgung von Sensoren verfügbar.

Socket U0: Pin1 = Masse, Pin2 = U0, Pin3 = +5V

Socket U1: Pin4 = Masse, Pin5 = U1, Pin6 = +5V

Socket U2: Pin1 = Masse, Pin2 = U2, Pin3 = +5V

Socket U3: Pin4 = Masse, Pin5 = U3, Pin6 = +5V

Socket U4: Pin1 = Masse, Pin2 = U4, Pin3 = +5V

Socket U5: Pin4 = Masse, Pin5 = U5, Pin6 = +5V

Socket U6: Pin1 = Masse, Pin2 = U6, Pin3 = +5V

Socket U7: Pin4 = Masse, Pin5 = U7, Pin6 = +5V

Alle analogen Eingänge und digitalen I/O sind ausgelegt für einen Spannungsbereich von 0 bis +5 Volt. Höhere oder negative Spannungen können zu Defekten an der Sensorbox führen. Um Schäden durch Überspannung zu vermeiden ist es sinnvoll, angeschlossene Sensoren und Sensorverstärker aus der Betriebsspannung der Sensorbox (+5V Pins) zu speisen unter Verwendung der Masseleitungen aus der Sensorbox.

Analogeingänge U0 ... U7

Der Analog / Digital- Wandler arbeitet mit ratiometrischer Referenz, d.h. misst die Eingangsspannung im Verhältnis zur Betriebsspannung. Für die meisten Sensorschaltungen ist dieses Messverfahren einfacher und genauer als eine konstante Referenz. Bei der Messung absoluter Spannungswerte ist aber mit einem gewissen Fehler zu rechnen.

Der **Eingangswiderstand** der Analogeingänge beträgt etwa 330 Kiloohm. Ein offener, nicht angeschlossener Eingang zeigt normalerweise die Eingangsspannung 0 an, übernimmt jedoch einen kleinen Bruchteil der an einem Nebeneingang anliegenden Spannung. Dieses Phänomen verschwindet, wenn man einen Sensor an diesen Ausgang anschliesst.

Kombinierte Digital I/O K0 ... K9,KA,KB:

Neben den analogen Eingängen hat die Sensorbox **12 kombinierte digitale Ein / Ausgänge**. Zwecks flexibler Nutzbarkeit führen die Leitungen der Applikations-Buchse ohne Schutzschaltung direkt zum Mikrocontroller. **Überspannungen oder Kurzschlüsse zerstören unter ungünstigen Umständen den Mikrocontroller!!**

Für jeden digitalen Ein/Ausgang lassen sich per Software 4 unterschiedliche Zustände programmieren:

-- "**Ausgang LOW**" Dabei wird die Leitung durch einen eingebauten Transistor per MIDI-Befehl auf Masse-Potential gezogen.

-- "**Ausgang HIGH**" ist das Gegenstück zu Ausgang Low, hier wird die Leitung durch einen eingebauten Transistor auf das Potential der internen 5V Betriebsspannung gezogen.

Als "Ausgang Low" oder "Ausgang High" geschaltete Leitungen können nicht als Eingang dienen, obwohl der Zustand dieser Leitungen weiterhin abgefragt werden kann (es wird dann der jeweilige logische Ausgangszustand gemeldet)

Angeschlossene Sensoren sollten so konstruiert sein, dass ein per Software "Ausgang Low" geschalteter Ausgang nicht durch den Sensor auf "HIGH" gezogen werden kann --- und dass ein per Software "Ausgang High" geschalteter Ausgang nicht durch den Sensor auf "LOW" gezogen werden kann. Unter ungünstigen Umständen kann sonst der Sensor und/oder der Mikrocontroller zerstört werden.

-- "**Eingang hochohmig**" ("**hi Z**") ist ein Eingang ohne aktivierten Pull-Up Widerstand.

Der Vorteil dieser Schaltungsart ist die geringe Belastung angeschlossener Sensoren durch die Eingangsschaltung. Der Nachteil ist der in der Praxis undefinierte Spannungswert bei offenem Eingang und ein etwas erhöhtes Risiko, den Eingang durch elektrostatische Aufladung zu zerstören.

-- "**Eingang HIGH**" Technisch gesehen wird dabei innerhalb der Box ein sog. **Pull-Up Widerstand** (einige 10 Kiloohm) von der Eingangsleitung nach +5Volt aktiviert.

Ohne extern anliegende Spannung meldet der Eingang deshalb immer eine "1". Wird der Eingang durch einen externen Einfluss auf "LOW"-Potential gezogen, so wird entweder automatisch eine Meldung abgegeben oder der Zustand kann abgefragt werden ("Polling"). Eine auf "Eingang mit Pull-Up Widerstand" geschaltete Leitung kann auch als relativ kurzschlussfester Ausgang dienen, allerdings keine hohe Last ansteuern.

Externe Lasten wie Relais oder Leuchtdioden sollten nach Möglichkeit zwischen Digital I/O und +5 Volt geschaltet werden anstatt zwischen Digital I/O und Masse, weil der "Ausgang Low" Zustand einen kräftigeren Transistor aufweist. Der maximal aus einem Ausgang entnehmbare Strom beträgt 10 mA. Bei induktiven Lasten wie z.B. Relais oder Motoren prinzipbedingt auftretende Spannungsspitzen müssen unbedingt mit einer "Löschdiode" vernichtet werden. Eine Schottky-Diode bietet den besten Schutz.

Alternative Funktionen der digitalen I/O:

Um mit der Sensorbox Steuer- und Regel- Anwendungen mit Schritt- und Servo-Motoren zu realisieren, **ist Digital I/O Nr.0 als Frequenzgenerator G konfigurierbar**. Sowohl die Ausgangsfrequenz (ca.0,7 Hz bis ca. 23 kHz) als auch die Dauer der positiven Impulsphase können unabhängig voneinander programmiert werden.

Digital I/O Nr.0 und Nr.2 sind konfigurierbar als Pulsweitemodulator W0 bzw. W2 mit ca. 5,9 kHz Grundfrequenz (10 Bit Auflösung) bzw. ca. 23,5 kHz (bei 8 Bit Auflösung). Das digitale pulsweitemodulierte Signal kann z.B. zur effizienten Drehzahlsteuerung eines Gleichstrom-Elektromotors oder zur Ansteuerung eines Fernsteuer-Servo benutzt werden. W2 kann allerdings nur gleichzeitig mit W0 und mit gleicher Grundfrequenz als Pulsweitemodulator aktiviert werden.

Digital I/O Nr. 6 kann ebenfalls als Pulsweitemodulator W6 aktiviert werden. Allerdings beträgt die Auflösung nur 8 Bit. Für diesen Pulsweitemodulator ist bereits ein Tiefpassfilter und Verstärker eingebaut, mit dem das pulsweitemodulierte Digitalsignal in eine den Tastverhältnis proportionale Gleichspannung 0 bis 10 Volt umgewandelt wird. Diese einstellbare Spannung kann alternativ auf den Ausgang von Digital I/O Nr. 6 gelegt werden:

Dazu Gehäusedeckel abnehmen (4 Kreuzschlitz-Schrauben). Neben dem Mikrocontroller-IC befindet sich ein 3-poliger Jumper. Im Lieferzustand ist K6 als normaler Kombi I/O konfiguriert, der Jumper steckt in Richtung zur Beschriftung des Mikrocontrollers. Wird der Jumper umgesteckt in Richtung zum 8-poligen Steckverbinder, wird die variable analoge Ausgangsspannung 0 bis 10 Volt auf Ausgang K6 gelegt. Dieser ist dann nicht mehr als Eingang nutzbar. **Wenn der optionale 0-10 Volt Ausgang aktiviert ist, muss unbedingt darauf geachtet werden, dass diese Spannung mit keinem anderen Eingang verbunden wird - dann besteht ein grosses Risiko, den eingebauten Mikrocontroller zu zerstören.**

Digital I/O Nr. 1, 3, 7 und 9 sind intern jeweils mit Impulszählern (J1/X1, J3/X3, J7/X7, J9/X9) gekoppelt, die jeweils mit jeder negativen Flanke am Digital I/O weiterzählen. Während J1, J3, J7 und J9 stets aufwärts zählen, **wird bei X1, X3, X7 und X9 die Zählrichtung bestimmt durch den logischen Pegel des benachbarten Eingangs, also Nr. 2, 4, 8 und A:**

-- ist dessen Pegel "High", wird aufwärts gezählt.

Nach "dez65535 = hexFFFF" folgt Zählerstand "0".

-- ist dessen Pegel "Low", wird abwärts gezählt.

Nach "0" folgt der Zählerstand"dez65535 = hexFFFF".

Mit Hilfe dieser Technik eignen sich diese Digital I/O Paare vor allem zur Auswertung von Rotations-Encodern mit Richtungserkennung durch 2-Phasen-Takt. Bei einfachen mechanischen Encodern ist eine Entprellung durch externe RC-Glieder notwendig, deren Dimensionierung sich nach der vorgesehenen Drehgeschwindigkeit richtet.

Alle Zähler werten die negativen Flanken an den entsprechenden Digital I/O im Hintergrund aus und schränken die elementaren I/O Funktionen in keiner Weise ein. Es wird der am I/O Anschluss physikalisch anliegende Spannungswert ausgewertet, d.h. die Zähler registrieren sowohl Eingangsimpulse als auch ausgegebene Impulse. D.h. wenn Pulsweitemodulator W2 aktiv ist, kann X1 nicht ausgewertet werden.

Installation und Inbetriebnahme

Sensoren und Aktuatoren sollten erst nach einer erfolgreichen Inbetriebnahme der Kommando-Schnittstelle angeschlossen werden.

Zuerst werden die MIDI-Verbindungen wie üblich gesteckt: MIDI OUT des Steuergeräts nach MIDI IN der MIDI Sensorbox und MIDI OUT der Sensorbox nach MIDI IN des Steuergeräts. Schließlich wird mit dem Codierschalter der zu benutzende MIDI-Kanal voreingestellt.

Wir empfehlen, zum ersten Kennenlernen den Codierschalter auf "1" zu stellen.

Die MIDI Sensorbox kann komplett gesteuert werden mit **MIDI Kanalnachrichten** sowie mit **System-Exklusiven Befehlen**, die eine einfache ASCII textbasierte Syntax einbetten. Beide Befehlstypen können beliebig gemischt werden.

Nachdem diese Vorbereitungen getroffen sind, wird die **Stromversorgung** der Sensorbox angeschlossen und eingeschaltet.

An das angeschlossene MIDI-Equipment wird beim Einschalten als Startmeldung eine SysEx Nachricht geschickt, die etwa folgenden Text enthält: "Sensorbox v5.14". Normalerweise muss die LED an der Frontplatte jetzt grün leuchten. Alle digitalen I/O Leitungen werden in der Grundkonfiguration als Eingänge mit Pull-Up Widerstand initialisiert, d.h. melden beim Abfragen einen logischen High-Zustand. Die Analogeingänge hingegen werden stets als hochohmige Eingänge initialisiert, haben jedoch hardwareseitig einen 330 Kiloohm Widerstand nach Masse eingebaut. Sie melden beim Abfragen ohne angeschlossene Sensoren den Wert 0 oder eine sehr geringe positive Eingangsspannung.

Wenn ein erstes Gefühl für die Handhabung der Sensorbox vorhanden ist, werden **Sensoren oder Aktuatoren einer nach dem anderen an die analogen bzw. digitalen I/O** angeschlossen und sofort ausprobiert. Wegen der Vielfalt der in Frage kommenden Komponenten kann hier keine allgemein gültige Anleitung gegeben werden. Da bei einem falschen Anschluss immer ein Zerstörungsrisiko sowohl für die angeschlossenen Komponenten als auch für den Mikrocontroller besteht, **sollte jede Funktion unmittelbar nach dem Anschluss getestet und im Zweifelsfall sofort die Stromversorgung abgeschaltet werden.** Dabei ist die interaktive Bedienung mit einem möglichst vielseitigen MIDI-Monitor sehr hilfreich.

Elementare Befehle zum schnellen Ausprobieren:

Digitale I/O mit CHANNEL PRESSURE konfigurieren

Das Datenbyte des CHANNEL PRESSURE Befehls setzt sich zusammen aus der I/O Nummer (0 bis 11=hexB) plus einer konstanten Zahl, mit der der gewünschte I/O Zustand beschrieben wird.

I/O Nummer + 0 setzt die Leitung als Ausgang mit LOW-Pegel.

I/O Nummer + 16 (hex 10) setzt die Leitung als Ausgang mit HIGH-Pegel.

I/O Nummer + 32 (hex 20) konfiguriert einen hochohmigen Eingang

I/O Nummer + 48 (hex 30) konfiguriert einen Eingang mit Pull-Up Widerstand

Den digitalen Zustand einer I/O Leitung auslesen („pollen“) mit CHANNEL PRESSURE:

Dazu wird das Datenbyte eines CHANNEL PRESSURE Befehls gleich

I/O Nummer + 64 (hex 40) gesetzt. (I/O Nummer = 0 bis 11=hexB)

Als Antwort kommt eine CONTROL CHANGE Nachricht (Konfiguration bei Lieferung, Details siehe unten).

Die Spannung an einem Analogeingang auslesen ("pollen") mit CHANNEL PRESSURE:

Dazu wird das Datenbyte eines CHANNEL PRESSURE Befehls gleich

I/O Nummer + 80 (hex 50) gesetzt. (I/O Nummer = 0 bis 7)

Als Antwort kommt eine PITCH CHANGE Nachricht. Details siehe unten..

Jeder Zugriff auf die Sensorbox kann sowohl mit MIDI-Kalnnachrichten als auch mit MIDI System-Exklusiven Nachrichten (nachfolgend als SysEx bezeichnet) durchgeführt werden. Ggf. darauf von der Sensorbox zurückgeschickte Antwort-Meldungen kommen jeweils in dem Format, in dem die Anfrage formuliert war. Automatische Meldungen werden als MIDI-Kalnnachrichten gesendet (Standard bei Auslieferung) oder per Befehl einstellbar als SysEx Nachricht.

MIDI-Kommunikationsprotokoll

der MIDI - Sensorbox (MIDI-Kanäle 1 bis 12=hexC)

Bekanntlich ist der in die MIDI-Statusbytes encodierte Kanalwert um eins kleiner als der am Codierschalter eingestellte "offizielle" MIDI Kanal, nimmt also Werte von 0 bis 11 an. Dies ist beim Schreiben eigener Steuerprogramme zu beachten !

Der Inhalt der Nachrichten, wie z.B. Notenwerte und Controller-Nummern, orientiert sich nicht an den für Musikanwendungen üblichen Einteilungen, sondern ist für kompakte und eindeutige Steuerung der Sensorbox optimiert. Daher können Konflikte auftreten, wenn dieses Equipment auf einer MIDI-Leitung direkt mit Musikanwendungen zusammenarbeitet. Beseitigung durch entsprechende Wahl der MIDI-Kanäle.

Übersicht aller MIDI Kanalnachrichten-Befehle (MIDI Implementation Chart)

(anwendbar nur in den Codierschalter-Stellungen 1 bis C)

Dabei werden folgende **Abkürzungen** verwendet:

DB heisst "Datenbyte",

DB1 heisst "1.Datenbyte" (z.B. Controller Nummer), DB2 heisst "2.Datenbyte" (z.B. Controller Wert).

Der im Befehl codierte MIDI- Kanal ist gleich der Stellung des Codierschalters.

Bei PROGRAM CHANGE Befehlen ist zu beachten, dass die meisten MIDI-Geräte und -Programme **bei "Programm 1" das Datenbyte 0 senden usw.** In der Tabelle bezeichnet "DB" grundsätzlich das physikalisch übertragene Datenbyte.

MIDI-Nachricht	typ. Datenbytes	Befehl senden	Antwort / Meldung
NOTE OFF		nicht implementiert	nicht implementiert
NOTE ON MIDI-Kanal wie Codierschalter	Bits 5 und 6 bestimmen die Art der Modifikation	Digital I/O kollektiv einstellen (nur Bits=1 modifizieren, die anderen unverändert lassen)	
	DB1 = 0 0 0 KB KA K9 K8 K7 DB2 = 0 K6 K5 K4 K3 K2 K1 K0	gesetzte Bits (=1) schalten auf AUSGANG LOW Bits = 0 bleiben unverändert	
	DB1 = 0 0 1 KB KA K9 K8 K7 DB2 = 0 K6 K5 K4 K3 K2 K1 K0	gesetzte Bits (=1) schalten auf AUSGANG HIGH Bits = 0 bleiben unverändert	
	DB1 = 0 1 0 KB KA K9 K8 K7 DB2 = 0 K6 K5 K4 K3 K2 K1 K0	gesetzte Bits (=1) schalten auf HOCHOHMIGEN EINGANG Bits = 0 bleiben unverändert	
	DB1 = 0 1 1 KB KA K9 K8 K7 DB2 = 0 K6 K5 K4 K3 K2 K1 K0	gesetzte Bits (=1) schalten auf EINGANG MIT PULL-UP- Widerstand Bits = 0 bleiben unverändert	
			Kollektive Ausgabe aller digitalen I/O: DB1 =0 0 KB KA K9 K8 K7 K6 DB2 = 0 0 K5 K4 K3 K2 K1 K0
POLY KEY PRESSURE MIDI-Kanal wie Codierschalter	DB1 = u	Automat.Nachrichten für analog IN Nr. u Schwelle= DB2	
	DB1 = 16 + u (hex 10+ u)	Automat.Nachrichten für analog IN Nr.u Schwelle= DB2+128	
	DB1 = 64 (hex 40)	DB2 = Zeitkonstante für Entprellen digitaler I/O, (1 bis 127 Millisekunden, s.Text)	

MIDI-Nachricht	typ. Datenbytes	Befehl senden	Antwort / Meldung
POLY KEY PRESSURE MIDI-Kanal wie Codierschalter	DB1= 112 (hex 70)	DB2= GerätelD (2 mal !)	
	DB1 = 127 DB2 = 80	Zähler Nr.1 zurücksetzen	
	DB1 = 127 DB2 = 81	Impuls-Zähler Nr.1 lesen	
	DB1 = 127 DB2 = 82	Encoder-Zähler Nr.1 lesen	
	DB1 = 127 DB2 = 83	Zähler Nr.3 zurücksetzen	
	DB1 = 127 DB2 = 84	Impuls-Zähler Nr.3 lesen	
	DB1 = 127 DB2 = 85	Encoder-Zähler Nr.3 lesen	
	DB1 = 127 DB2 = 86	Zähler Nr.7 zurücksetzen	
	DB1 = 127 DB2 = 87	Impuls-Zähler Nr.7 lesen	
	DB1 = 127 DB2 = 88	Encoder-Zähler Nr.7 lesen	
	DB1 = 127 DB2 = 89	Zähler Nr.9 zurücksetzen	
	DB1 = 127 DB2 = 90	Impuls-Zähler Nr.9 lesen	
	DB1 = 127 DB2 = 91	Encoder-Zähler Nr.9 lesen	
POLY KEY PRESSURE MIDI-Kanal wie Codierschalter			Zählerstand melden 14 Bit Zählerkapazität DB1 = 7 LSB (Bits 6,5,4,3,2,1,0) DB2 = 7 MSB (Bits 13,12,11,10,9,8,7) Keine Unterscheidung, von welchem Zähler die Meldung
CONTROL CHANGE MIDI-Kanal wie Codierschalter	DB1 = I/O Nummer I/O Nr. 0 codieren als 16 (hex 10) ! Erläuterungen siehe Text !	DB2 = Befehlstyp DB2 = 0 Ausgang LOW DB2 = 1 Ausgang LOW DB2 = 2 Eingang hochohmig DB2 = 3 Eingang mit Pullup R DB2 = 4 Digital IO pollen DB2 = 5 Analogeing. pollen DB2 = 6 Automat.Nachr. AUS DB2 = 7 Automat.Nachr. EIN	
			Meldung einzelner dig. I/O: DB1 = I/O Nummer DB2 = 0 oder 1
PROGRAM CHANGE MIDI-Kanal wie Codierschalter	DB = 0	Automat. Meldungen digitaler I/O individuell codieren als C-CHANGE od.CHAN-PRESS	Fehlermeldungen kommen als PROGRAM CHANGE Nachrichten, DB=0
	DB = 1	Automat. Meldungen digitaler I/O kollektiv als NOTE ON	
	DB = 16 (hex 10)	Automatische Nachrichten als MIDI Kanalnachrichten	
	DB = 17 (hex 11)	Automatische Nachrichten als SysEx formatierter Text	
	DB = 32 (hex 20)	Zustandsmeldungen der digitalen I/O senden als CONTROL CHANGE	
	DB = 33 (hex 21)	Zustandsmeldungen der digitalen I/O senden als CHANNEL PRESSURE	
	DB = 48 (hex 30)	Zustandsmeldungen OHNE Running State senden	
	DB = 49 (hex 31)	dto MIT Running State senden	
	DB = 64 (hex 40)	Running State in Befehlen NICHT akzeptieren	
	DB = 65 (hex 41)	Running State in Befehlen AKZEPTIEREN	

MIDI-Nachricht	typ. Datenbytes	Befehl senden	Antwort / Meldung
PROGRAM CHANGE MIDI-Kanal wie Codierschalter	DB = 96 (hex 60)	Preset speichern	
	DB = 112 (hex 70)	Preset laden und I/O Konfig. entsprechend einstellen	
	DB = 125 (hex 7D)	gelöschtes Preset reaktivieren	
	DB = 124 (hex 7E)	Preset und I/O Konfiguration in Lieferzustand versetzen	
	DB = 127 (hex 7F)	I/O Konfiguration auf Lieferzustand initialisieren, Preset bleibt unverändert	
CHANNEL PRESSURE	DB = 0mmm kkkk	kkkk = I/O Nr. 0 bis 11	
MIDI-Kanal wie	mmm = 0 (kkkk)	digit. Ausgang Low	
Codierschalter	mmm = 1 (kkkk + 16)	digit. Ausgang High	
	mmm = 2 (kkkk + 32)	digit. Eingang hochohmig	
	mmm = 3 (kkkk + 48)	digit. Eingang mit Pull-Up Wid.	
	mmm = 4 (kkkk + 64)	digital-I/O pollen	
	mmm = 5 (kkkk + 80)	analog. Eingang pollen	
	mmm = 6 (kkkk + 96)	digit. autom.Nachr. AUS	
	mmm = 7 (kkkk + 112)	digit. autom.Nachr. EIN	
			Meldungen einzelner digitaler I/O: DB = 0 0 0 X k k k k X = 0 oder 1 , je nachdem ob der logische Pegel der I/O Leitung High oder Low ist. Anders formuliert: DB = I/O Nummer wenn Low DB = I/O Nummer + 16 wenn High
PITCH CHANGE MIDI-Kanal wie Codierschalter	DB1= hhhhh Bei geringer Genauig. reicht DB1= 0	Frequenzgenerator I/O 0 High-Periode = DB2 * 32 + hhhhh	
	DB1= 32 + TTTTT Bei geringer Genauig. reicht DB1= 32 (hex20)	Frequenzgenerator I/O 0 Gesamtperiode = DB2 * 32 + TTTTT	
	DB1= 64 + MM+ ppp Bei geringer Genauig. reicht DB1= 64 (hex40)	Start Pulsweitenumodulator I/O 0 High-Phase = DB2 * 8 + ppp MM siehe Text	
	DB1= 96 + ppp Bei geringer Genauig. reicht DB1= 96 (hex60)	Start Pulsweitenumodulator I/O 2 High-Phase = DB2 * 8 + ppp MM immer wie PWM0	
	DB1= 112 + MMM +p Bei geringer Genauig. reicht DB1= 112 (hex70)	Start Pulsweitenumodulator I/O 6 High-Phase = DB2 * 2 + p MMM siehe Text	
			Meldung des aktuellen Werts am Analogeingang U DB1 = 0 u u u n n n n DB2 = 0 U U U U U U U U nnnn = Eingangsnummer UUUUUUUUuuuu = Messwert Die LSB uuu in DB1 können ggf. vernachlässigt werden.
SYSTEM EXCLUSIVE	kanal-unabhängig	kapselt alle ASCII-Befehle für MIDI	kapselt alle ASCII-Befehle für MIDI
MIDI RESET		Reset, lädt Preset	Reset, lädt Preset

I/O Leitungen individuell einstellen oder abfragen mit:

CONTROL CHANGE

MIDI-Kanal wie Codierschalter

1. Datenbyte = kkkk = zu behandelnde I/O Nummer (0 bis 11)

Ausnahme: I/O Nr. 0 muss mit 16 (hex10) codiert werden,

weil sonst ein Konflikt mit der Filterung von Bank Select Befehlen entsteht.

Siehe Anmerkung zu "Presets laden mit PROGRAM CHANGE", weiter unten

2. Datenbyte = mmm beschreibt die auszuführende Aktion:

mmm = 0 I/O Nr. kkkk als Ausgang Low schalten

mmm = 1 I/O Nr. kkkk als Ausgang High schalten

mmm = 2 I/O Nr. kkkk als hochohmigen Eingang schalten

Hochohmige Eingänge sind sehr stöempfindlich und haben häufig einen unstabilen logischen Zustand. Daher Eingänge nur bei Anschluss entsprechender Sensoren o.ä. hochohmig einstellen!

mmm = 3 I/O Nr. kkkk als Eingang mit Pull-Up Widerstand schalten

Dabei wird in der Sensorbox ein Widerstand von einigen 10 Kiloohm vom Eingang nach 5+Volt geschaltet. Beim Anschluss von Schaltern oder manchen Sensortypen erspart dies den externen Anschluss von Last-Widerständen. Unbenutzte digitale I/O sind mit Pull-Up Widerstand auch am wenigsten störanfällig, daher sind die digitalen I/O bei Lieferung so konfiguriert. Sie können auch als Ausgang mit kurzschluss-sicherem "weichem" High-Pegel eingesetzt (dies ist der Standard bei den bekannten 8051 Mikrocontrollern).

mmm = 4 digitalen Zustand des I/O Nr. kkkk abfragen ("pollen")

Die Antwort kommt je nach Voreinstellung (mit PROGRAM CHANGE) als:

--- **CONTROL CHANGE** Nachricht (default Einstellung), wobei das erste Datenbyte (Controller Nummer) = kkkk ist und das zweite Datenbyte (Controller Wert) entsprechend dem Leitungszustand gleich 0 oder 1 ist

--- **CHANNEL PRESSURE** Nachricht. Dann enthalten die niederwertigen Bits 0-3 die I/O Nummer kkkk und das 4.Bit ist je nach gelesenen logischen Pegel der Leitung 0 oder 1.

Anders formuliert: befindet sich der Eingang auf logischem Low-Pegel, dann kommt die Eingangsnummer zurück. Befindet sich der Eingang auf logischem High-Pegel, dann kommt die Eingangsnummer + 16 zurück.

Sonderfall: 1. Datenbyte kkkk=13 (hexD):

Dann werden die digitalen Zustände **aller** I/O -Leitungen als **NOTE ON** Nachricht übertragen. Die 6 höchstwertigen Bits (digital I/O 11=B,10=A,9,8,7,6) werden im 1. Datenbyte übertragen , die 6 niederwertigen Bits (Analogeingänge 5 bis 0) werden im 2. Datenbyte übertragen.

hex9n 00d_Bd_Ad₉d₈d₇d₆ 00d₅d₄d₃d₂d₁d₀

mmm = 5 Spannung am analogen Eingang Nr. kkkk (0 bis 7) abfragen

Die **Antwort** kommt als **PITCH CHANGE Nachricht** im Format

0xE_n 0uuu kkkk 0UUU UUUU

wobei die "most significant" Bits UUU UUUU ins zweite Datenbyte (Velocity) und die 3 "least significant" Bits uuu ins erste Datenbyte (Notenwert) gepackt werden. Bei geringen Anforderungen an die Genauigkeit wird empfohlen, nur das 2. Datenbyte auszuwerten.

mmm = 6 automat. Zustandsmeldungen des digitalen I/O Nr. kkk **AUS**schalten

mmm = 7 automat. Zustandsmeldungen des digitalen I/O Nr. kkkk **EIN**schalten

Anmerkung: je nach Voreinstellung (mit PROGRAM CHANGE Befehl) werden automatische Nachrichten der digitalen I/O entweder individuell pro I/O gesendet (CONTROL CHANGE oder CHANNEL PRESSURE) oder als kollektive Meldung aller I/O (NOTE ON). Das Datenformat automatischer Meldungen ist gleich dem oben unter "Abfragen" beschriebenen.

Alternativ:

I/O Leitungen individuell einstellen oder abfragen mit:

CHANNEL PRESSURE

MIDI-Kanal wie Codierschalter

Datenbyte = **0mmm kkkk** bzw. **M + kkkk**

Dabei ist kkkk die zu behandelnde I/O Nummer (0 bis 11)

und mmm bzw M (=mmm * 16) beschreibt die auszuführende Aktion:

mmm = 0 (M=0): I/O Nr. kkkk als Ausgang Low schalten

mmm = 1 (M=16 / hex10) I/O Nr. kkkk als Ausgang High schalten

mmm = 2 (M=32 / hex20) I/O Nr. kkkk als hochohmigen Eingang schalten

Hochohmige Eingänge sind sehr stöempfindlich und haben häufig einen unstabilen logischen Zustand. Daher Eingänge nur bei Anschluss entsprechender Sensoren o.ä. hochohmig einstellen!

mmm = 3 (M=48 / hex30): I/O Nr. kkkk als Eingang mit Pull-Up Widerst. schalten

Dabei wird in der Sensorbox ein Widerstand von einigen 10 Kiloohm vom Eingang nach 5+Volt geschaltet. Beim Anschluss von Schaltern oder manchen Sensortypen erspart dies den externen Anschluss von Last-Widerständen. Unbenutzte digitale I/O sind mit Pull-Up Widerstand auch am wenigsten störanfällig, daher sind die digitalen I/O bei Lieferung so konfiguriert. Sie können auch als Ausgang mit kurzschluss-sicherem "weichem" High-Pegel wirken (dies ist der Standard bei den bekannten 8051 Mikrocontrollern).

mmm = 4 (M=64 / hex40) digitalen Zustand des I/O Nr. kkkk abfragen ("pollen")

Die Antwort kommt je nach Voreinstellung (mit PROGRAM CHANGE) als:

--- **CONTROL CHANGE** Nachricht (default Einstellung), wobei das erste Datenbyte (Controller Nummer) = kkkk ist und das zweite Datenbyte (Controller Wert) entsprechend dem Leitungszustand gleich 0 oder 1 ist ... oder als

--- **CHANNEL PRESSURE** Nachricht. Dann enthalten die niederwertigen Bits 0-3 die I/O Nummer kkkk und das 4.Bit ist je nach gelesenen logischen Pegel der Leitung 0 oder 1. Anders formuliert: befindet sich der Eingang auf logischem Low-Pegel, dann kommt die Eingangsnummer zurück. Befindet sich der Eingang auf logischem High-Pegel, dann kommt die Eingangsnummer + 16 zurück.

Sonderfall: kkkk=13 (hexD): (d.h. Datenbyte = hex 4D = dez. 77)

Dann werden die digitalen Zustände **aller** I/O -Leitungen als **NOTE ON** Nachricht übertragen. Die 6 höchstwertigen Bits (digital I/O 11=B,10=A,9,8,7,6) werden im 1. Datenbyte übertragen , die 6 niederwertigen Bits (Analogeingänge 5 bis 0) werden im 2. Datenbyte übertragen.

hex9n 00d_Bd_Ad₉d₈d₇d₆ 00d₅d₄d₃d₂d₁d₀

mmm = 5 (M=80 / hex50) Spannung am analogen Eingang Nr. kkkk (0 bis 7) abfragen

Die Antwort kommt als **PITCH CHANGE** Nachricht im Format

0xE_n 0uuu kkkk 0UUU UUUU

wobei die "most significant" Bits UUU UUUU ins zweite Datenbyte (Velocity) und die 3 "least significant" Bits uuu ins erste Datenbyte (Notenwert) gepackt werden. Bei geringen Anforderungen an die Genauigkeit wird empfohlen, nur das 2. Datenbyte auszuwerten.

mmm = 6 (M=96 / hex60) automat. Zustandsmeldungen

des digitalen I/O Nr. kkkk **AUS**schalten

mmm = 7 (M=112 / hex70) automat. Zustandsmeldungen

des digitalen I/O Nr. kkkk **EIN**schalten

Anmerkung: je nach Voreinstellung (mit PROGRAM CHANGE Befehl) werden automatische Nachrichten der digitalen I/O entweder individuell pro I/O gesendet (CONTROL CHANGE oder CHANNEL PRESSURE) oder als kollektive Meldung aller I/O (NOTE ON). Das Datenformat automatischer Meldungen ist gleich dem oben unter "Abfragen" beschriebenen.

I/O Leitungen kollektiv einstellen mit:

NOTE ON

MIDI-Kanal wie Codierschalter

1. Datenbyte = **Aktionstyp MM** und 5 "most significant Bits" **0 M M K_B K_A K₉ K₈ K₇**
2. Datenbyte = 7 "Least significant bits" **0 K₆ K₅ K₄ K₃ K₂ K₁ K₀**

Um eine differenzierte Modifikation aller I/O Leitungen zu ermöglichen, **wirkt dieser Befehl immer und nur auf diejenigen I/O Leitungen, deren entsprechendes Bit gesetzt (=1) ist.** Die Leitungen, deren entsprechendes Bit nicht gesetzt ist (=0), werden nicht beeinflusst.
Mit dem Aktionstyp MM wird angegeben, welche Aktion ausgeführt werden soll:

MM = 00 (d.h. nichts zur Bitkombination der I/O addieren)

die I/O Leitungen deren Bits gesetzt sind, werden als **Ausgang LOW** geschaltet

MM = 01 (d.h. **32 = hex20** zur Bitkombination der I/O addieren)

die I/O Leitungen deren Bits gesetzt sind, werden als **Ausgang HIGH** geschaltet

MM = 10 (d.h. **64 = hex40** zur Bitkombination der I/O addieren)

die I/O Leitungen deren Bits gesetzt sind, werden als **hochohmiger Eingang** geschaltet

MM = 11 (d.h. **96 = hex60** zur Bitkombination der I/O addieren)

die I/O Leitungen deren Bits gesetzt sind, werden als **Eingang mit Pull-Up Widerstand** geschaltet

automatische Nachrichten der Analogeingänge einstellen mit :

POLY KEY PRESSURE (=POLYPHONIC AFTERTOUCHE)

MIDI-Kanal wie Codierschalter

1 Datenbyte = **Eingangs-Nr** (0 bis 7)

2. Datenbyte = Schwelle der Meldungen (0 bis 127)

Sobald der aktuelle digitalisierte Wert der Eingangsspannung von der vorherigen Meldung mindestens um die Schwelle abweicht, wird eine **NOTE ON Nachricht** gesendet. Das Datenformat ist das gleiche wie bei abgefragten ("gepollten") Nachrichten, siehe oben bei CHANNEL PRESSURE Befehlen.

Wenn das 2. Datenbyte = 0 ist, werden vorher aktivierte automatische Meldungen dieses Eingangs AUSgeschaltet.

Soll die Schwelle höher als 127 eingestellt werden, so lautet der Befehl

1 Datenbyte = **Eingangs-Nr** (0 bis 7) + **16 (hex 10)**

2. Datenbyte = Schwelle der Meldungen **minus 128**

d.h. die Schwelle berechnet sich: 2. Datenbyte plus 128

Auf diese Art lassen sich Schwellen von 1 bis 255 einstellen, d.h. bis maximal 1/4 des Wertebereichs der digitalisierten analogen Spannungen.

Impulszähler 1 und Encoder-Zähler 1 zurücksetzen mit:

POLY KEY PRESSURE

MIDI-Kanal wie Codierschalter

1 Datenbyte = **127 (hex 7F)**

2. Datenbyte = **80 (hex 50)**

Impulszähler 3 und Encoder-Zähler 3 zurücksetzen: 2. Datenbyte = **83 (hex 53)**

Impulszähler 7 und Encoder-Zähler 7 zurücksetzen: 2. Datenbyte = **86 (hex 56)**

Impulszähler 9 und Encoder-Zähler 9 zurücksetzen: 2. Datenbyte = **89 (hex 59)**

Impulszähler 1 (I/O Nr. 1) auslesen mit:

POLY KEY PRESSURE

MIDI-Kanal wie Codierschalter

1 Datenbyte = **127 (hex 7F)**

2. Datenbyte = **81 (hex 51)**

Impulszähler 3 auslesen: 2. Datenbyte = **84 (hex 54)**

Impulszähler 7 auslesen: 2. Datenbyte = **87 (hex 57)**

Impulszähler 9 auslesen: 2. Datenbyte = **90 (hex 5A)**

Bei allen Zähler-Auslesungen kommt die **Antwort als POLY KEY PRESSURE Nachricht im 14 Bit Format**. Die 7 höchstwertigen Bits werden in das 2. Datenbyte gepackt und die 7 niederwertigen Bits in das 1. Datenbyte. Wegen des eingeschränkten Datenumfangs einer MIDI-Kanalnachricht ist der maximale Zählerwert kleiner als bei ASCII-Befehlen. Die Zähler laufen zyklisch über.

Achtung: alle Zähler-Abfragen erzeugen das gleiche Datenformat. Um Verwechslungen zu vermeiden, muss die Antwort ausgewertet werden, bevor ggf. der nächste Zählerauslese-Befehl geschickt wird.

Encoder-Zähler (I/O Nr. 1, Richtungserkennung mit I/O Nr. 2) auslesen mit:

POLY KEY PRESSURE

MIDI-Kanal wie Codierschalter

1 Datenbyte = **127 (hex 7F)**

2. Datenbyte = **82 (hex 52)**

Encoder-Zähler 3 auslesen: 2. Datenbyte = **85 (hex 55)**

Encoder-Zähler 7 auslesen: 2. Datenbyte = **88 (hex 58)**

Encoder-Zähler 9 auslesen: 2. Datenbyte = **91 (hex 5B)**

Bei allen Zähler-Auslesungen kommt die **Antwort als POLY KEY PRESSURE Nachricht im 14 Bit Format**. Die 7 höchstwertigen Bits werden in das 2. Datenbyte gepackt und die 7 niederwertigen Bits in das 1. Datenbyte. Wegen des eingeschränkten Datenumfangs einer MIDI-Kanalnachricht ist der maximale Zählerwert kleiner als bei ASCII-Befehlen. Die Zähler laufen zyklisch über.

Achtung: alle Zähler-Abfragen erzeugen das gleiche Datenformat. Um Verwechslungen zu vermeiden, muss die Antwort ausgewertet werden, bevor ggf. der nächste Zählerauslese-Befehl geschickt wird.

Frequenzgenerator I/O Nr.0 High-Phase (vor-)einstellen mit :

PITCH CHANGE

MIDI-Kanal wie Codierschalter

1 Datenbyte = **000h hhhh** (h hhhh = 0 bis 31)

2.Datenbyte = **0HHH HHHH = (HighPhase) / 32, Divisionsrest = h hhhh**
(HHH HHHH = 0 bis 127)

d.h. die HighPhase berechnet sich zu (HHH HHHH x 32 + hhhh)

Bei geringeren Präzisionsansprüchen reicht es aus, das erste Datenbyte = 0 zu setzen. Dann gibt das zweite Datenbyte grob die Dauer der High-Phase an.

Frequenzgenerator I/O Nr. 0 Gesamtperiode einstellen und starten mit :

PITCH CHANGE

MIDI-Kanal wie Codierschalter

1 Datenbyte = **32 + t tttt** (t tttt = 0 bis 31)

2.Datenbyte = **0TTT TTTT = Gesamtperiode / 32, Divisionsrest = t tttt**
(TTT TTTT = 0 bis 127)

d.h. die Gesamtperiode berechnet sich zu (TTT TTTT x 32 + ttttt)

Bei geringeren Präzisionsansprüchen reicht es aus, das erste Datenbyte d.h. ttttt= 32 zu setzen. Dann gibt das zweite Datenbyte grob die Periodendauer an.

Um den Frequenzgenerator zu starten, muss zuvor die High-Phase passend eingetragen werden, sonst ist mit undefiniertem Verhalten zu rechnen. Bei laufendem Generator können beide Anteile unabhängig voneinander modifiziert werden, die Gesamtperiode muss jedoch stets größer sein als die High-Phase.

Periodendauer und High-Phase der Rechteckschwingung an I/O Nr.0 berechnen sich:

eingestellter Datenwert **mal ca. 21 Mikrosekunden.**

Wegen des eingeschränkten Datenumfangs einer MIDI-Kanalnachricht ist der Einstellbereich geringer als mit ASCII-Befehlen !

Ausschalten des Frequenzgenerators: Beliebigen Digitalzustand von I/O Nr.0 mit CONTROL CHANGE oder CHANNEL PRESSURE einstellen

Beispiele: (siehe auch Beispiele zum "W"-Befehl im 2.Teil "SysEx Befehle")

eine **Pitch Change** Nachricht mit 1.Datenbyte = 4 und 2.Datenbyte = 3
gefolgt von einer **Pitch Change** Nachricht mit 1.Datenbyte = 40 und 2.Datenbyte = 6
erzeugt eine Rechteckschwingung von ca.92 Hz mit 50% Tastverhältnis

eine **Pitch Change** Nachricht mit 1.Datenbyte = 4 und 2.Datenbyte = 1
gefolgt von einer **Pitch Change** Nachricht mit 1.Datenbyte = 63 und 2.Datenbyte = 127
erzeugt positive Impulse von 426 Mikrosekunden Dauer bei ca. 87 ms Gesamtperiode (=ca. 11Hz).

Pulsweitemodulator I/O Nr. 0 einstellen mit :

PITCH CHANGE

MIDI-Kanal wie Codierschalter

1 Datenbyte = **64 + MM + ppp** (ppp = 0 bis 7)

Durch Vorgabe von **MM ungleich 0** kann die Basisfrequenz beeinflusst werden.

Weitere Details siehe Anhang A

MM = 0: Basisfrequenz = 5865 Hz

MM = 8: Basisfrequenz = 733 Hz

MM = 16: Basisfrequenz = 92 Hz

MM = 24: Basisfrequenz = 23 Hz

2.Datenbyte = **0PPP PPPP = (High Phase) / 8 , Divisionsrest = ppp**
(PPP PPPP = 0 bis 127)

die HighPhase berechnet sich zu (PPP PPPP x 8 + ppp)

Die Auflösung beträgt bei Einstellung mit MIDI-Kanalbefehl immer 10 Bit, d.h. der High-Anteil der Schwingung ist = $100 \cdot (\text{HighPhase} / 1023)$ (in Prozent)

Ausschalten des Pulsweitemodulators: Beliebigen Digitalzustand von I/O Nr.0 mit CONTROL CHANGE, CHANNEL PRESSURE oder SysEx einstellen

Pulsweitemodulator I/O Nr. 2 einstellen mit :

PITCH CHANGE

MIDI-Kanal wie Codierschalter

1 Datenbyte = **96(hex60) + ppp** (ppp = 0 bis 7)

2. Datenbyte = **0PPP PPPP = (High Phase) / 8 , Divisionsrest = ppp**
(PPP PPPP = 0 bis 127)

die HighPhase berechnet sich zu (PPP PPPP x 8 + ppp)

Dieser Pulsweitemodulator teilt sich Ressourcen des Mikrocontrollers mit dem Frequenzgenerator/Pulsweitemodulator an I/O Nr.0. Daher kann er nur zusammen mit dem Pulsweitemodulator an I/O Nr.0 aktiv sein und mit derselben Grundfrequenz und Auflösung arbeiten.

Wird der Pulsweitemodulator an I/O Nr.0 ausgeschaltet, wird der Pulsweitemodulator an I/O Nr. 2 automatisch als Eingang mit Pull-Up Widerstand konfiguriert.

Wenn der Pulsweitemodulator an I/O Nr. 0 zwischen 8 und 10 Bit Auflösung wechselt, ändert sich das Tastverhältnis an I/O Nr.2

Wenn dieser Pulsweitemodulator aktiv ist, kann der Encoder-Zähler an I/O Nr.1 nicht ausgewertet werden.

Ausschalten des Pulsweitemodulators: Beliebigen Digitalzustand von I/O Nr.2 mit CONTROL CHANGE, CHANNEL PRESSURE oder SysEx einstellen oder die PWM an I/O Nr.0 de-aktivieren.

Pulsweitemodulator I/O Nr. 6 einstellen mit :

PITCH CHANGE

MIDI-Kanal wie Codierschalter

1 Datenbyte = **112(hex70) + MMM + p** (p = least significant Bit der High Phase =0 oder 1)

Durch Vorgabe von **MMM ungleich 0** kann die Basisfrequenz beeinflusst werden.

Weitere Details siehe Anhang A

MMM = 0: Basisfrequenz = 23529 Hz

MMM = 4: Basisfrequenz = 11765 Hz

MMM = 6: Basisfrequenz = 2941 Hz

MMM = 8: Basisfrequenz = 735 Hz

MMM = 10: Basisfrequenz = 368 Hz

MMM = 12: Basisfrequenz = 92 Hz

MMM = 14: Basisfrequenz = 23 Hz

2. Datenbyte = **0PPP PPPP = (HighPhase) / 2, Divisionsrest = p**
(PPP PPPP = 0 bis 127)

die HighPhase berechnet sich zu (PPP PPPP x 2 + p)

Die Auflösung dieses Pulsweite-Modulators beträgt immer 8 Bit, d.h. der High-Anteil der Schwingung ist = $100 \cdot (\text{HighPhase} / 255)$ (in Prozent)

Ausschalten des Pulsweitemodulators: Beliebigen Digitalzustand von I/O Nr.6 mit CONTROL CHANGE, CHANNEL PRESSURE oder SysEx einstellen

Betriebsart wechseln mit :

PROGRAM CHANGE

Die mit Datenbyte = 0 bis 65 vorgenommenen Einstellungen werden zusammen mit dem Preset fest abgespeichert und beim Laden des Presets reaktiviert.

MIDI-Kanal wie Codierschalter

Datenbyte = 0: Automatische Nachrichten der digitalen I/O werden individuell ausgegeben (default).

Datenbyte = 1: Automatische Nachrichten der digitalen I/O werden "kollektiv" als Zahl ausgegeben, welche die Zustände aller I/O repräsentiert (NOTE ON)

Datenbyte = 16 (hex10): Automatische Nachrichten als Kanalnachrichten senden (default).

Datenbyte = 17 (hex11): Automatische Nachrichten im SysEX Format senden.

Datenbyte = 32 (hex20): Zustandsmeldungen der digitalen I/O senden als CONTROL CHANGE (default)

Datenbyte = 33 (hex21): Zustandsmeldungen der digitalen I/O senden als CHANNEL PRESSURE

Datenbyte = 48 (hex30): Zustandsmeldungen **OHNE** Running State senden (default)

Datenbyte = 49 (hex31): Zustandsmeldungen **MIT** Running State senden

Datenbyte = 64 (hex40): Running State in Befehlen **NICHT** akzeptieren (default)

Datenbyte = 65 (hex41): Running State in Befehlen **AKZEPTIEREN**

Preset laden und speichern mit :

PROGRAM CHANGE

MIDI-Kanal wie Codierschalter

Datenbyte = 96 (hex 60): Preset **speichern**

Datenbyte = 112 (hex 90): Preset **laden**

Datenbyte = 125 (hex 7D): früher gelöscht Preset **reaktivieren**

Dieser Befehl geht davon aus, dass vorher ein gültiges Preset gespeichert wurde und danach die Box auf Grundeinstellungen zurückgesetzt wurde. Beim Rücksetzen auf Grundzustand bleiben die Details des Preset erhalten und werden nun wieder verfügbar gemacht.

Datenbyte = 126 (hex 7E): Preset und I/O Konfiguration auf Lieferstatus zurücksetzen

Datenbyte = 127 (hex 7F): **löscht** alle Puffer und Arbeitsregister.

D.h. der aktuelle Betriebszustand wird auf die Grundeinstellung zurückgesetzt. Das Preset wird jedoch nicht verändert.

Achtung: Bei vielen MIDI-Sequencern und anderen MIDI-Steuergeräten muss beim PROGRAM CHANGE Befehl die "logische" Programmnummer eins höher eingegeben werden, als sie "physikalisch" im Datenbyte codiert wird. (z.B. bei MAX Software und MIDIMAN Keyboards). In diesem Manual sind stets die "physikalisch" im Datenbyte zu codierenden Werte angegeben.

Viele MIDI-Geräte (z.B. programmierbare Keyboards, Sequencer) senden unmittelbar vor einem PROGRAM CHANGE Befehl automatisch einen BANK SELECT Befehl an Controller Nr. 0 (LSB) und

20 (MSB). Erhält die Sensorbox einen solcher Befehl **unmittelbar** gefolgt von einem PROGRAM CHANGE Befehl, wird der Controller-Befehl ignoriert.

SysEx Geräte-Identifikation verändern mit:

POLY KEY PRESSURE

MIDI-Kanal wie Codierschalter

1. Datenbyte = 112 (hex 70)
2. Datenbyte = neue GeräteID (0 bis 127)

Zur Sicherheit gegen unbeabsichtigtes Überschreiben muss dieser Befehl **zweimal unmittelbar nacheinander** eingegeben werden !

Datenformat MIDI Reset

Datenbyte = 255 (hex FF) MIDI System Reset

Wird dieser Befehl an die MIDI-Sensorbox gesendet, so erlischt die LED für ca. 1 Sekunde, dann wird das Gerät entspr. der Stellung des Codierschalters neu gestartet und ein vorhandenes Preset wird geladen.

System-Exclusive Befehle als gekapselter ASCII-Text

Grundsätzlicher Befehls-Aufbau:

<SysEx Header> Befehlscode Parameter [Befehlscode Parameter]....<EOX>

Der System-Exclusive Befehlsrahmen:

Der SysEx Rahmen umgibt die ASCII-Befehle als eine Art Klammer, um diese zum MIDI-Kontext kompatibel zu machen.

Der **SysEx-Header** besteht bei allen Befehlen aus folgender 5 Byte Sequenz, die neben dem Cinetix SysEx Manufacturer ID zusätzlich eine Geräte-Identifikationsnummer enthält:

Hexadezimal: **F0 00 20 5D GerätID**

bzw. Dezimal: **240 0 32 93 GerätID**

Mit Hilfe der Geräte-Identifikation können mehrere MIDI-Geräte von Cinetix auf einem MIDI-Bus unterschieden werden.

Bei Lieferung ist die GerätID = 0 eingestellt, aber sie kann einem POLY KEY PRESSURE Befehl (s.o.) abschalt-beständig verändert werden. Im Normalfall ist das jedoch nicht notwendig.

EOX ist ein einzelnes Byte, das bei MIDI immer den Wert **hex F7 = dezimal 247** hat.

In diesen System-Exclusive Befehlsrahmen können verschiedene ASCII-Befehle eingebettet bzw. gekapselt werden.

--- **Zuerst öffnet der SysEx Header den ASCII-Kommandointerpreter.**

--- **Ab nun wird jeder ASCII-Befehl ausgeführt, sobald die notwendige Anzahl von ASCII-Zeichen empfangen wurde.** Zahlen werden automatisch abgeschlossen, sobald die im jeweiligen Kontext maximal mögliche Anzahl ASCII-Stellen empfangen wurde. Bei kleineren Zahlen muss entweder einfach mit der Eingabe des nächsten Befehls begonnen werden oder die Zahl muss mit vorangestellten Nullen eingegeben werden oder es muss ein "EOX" eingegeben werden.

--- **Das EOX-Zeichen führt den letzten noch unbearbeiteten Befehl sofort aus und schließt den ASCII Kommandointerpreter. Zugleich wird damit wieder auf die Auswertung von Kanalnachrichten umgeschaltet** (bis zum nächsten SysEx Header).

Innerhalb eines System-Exclusive Befehlsrahmen bzw. -Paket können beliebig viele ASCII-Befehle gesendet werden, der Textinhalt pro SysEx Paket sollte jedoch 80 Bytes nicht überschreiten.

Nun zur Beschreibung der ASCII-Befehle als solche:

Jeder Steuerbefehl und jede Zustandsmeldung beginnt mit einem einzelnen **charakteristischen Buchstaben** (= Befehlscode), danach folgt, wenn notwendig, **eine Zahl als Wert-Parameter**.

--- Einige Befehlscodes dienen zur Auswahl bestimmter Input/Output-Methoden ("Devices" =virtuelle Geräte), das sind die Befehle U, K, S, J, M, W und G.

--- Andere Befehlscodes konkretisieren die mit dem zuvor selektierten Device auszuführende Aktion, das sind L, H, Z, P, R, Q, %, T und N.

--- Schliesslich gibt es eine dritte Art Befehlscodes, die das allgemeine

Systemverhalten definieren, dazu gehören z.B: X, Y, |, ~, @, & und \$.

Leerstellen, Komma und Gleichheitszeichen werden nicht interpretiert, d.h. können zur grafischen Auflockerung verwendet werden.

Alle Zeichen werden unabhängig von Groß- und Kleinschreibung interpretiert. Rückmeldungen kommen immer in Großbuchstaben.

Bei ungültigen Befehlen kommt als **Fehlermeldung** ein in einen SysEx Rahmen gepacktes Fragezeichen "?".

Elementares Befehlsformat für die häufigsten Operationen:

Als exemplarisches **Beispiel** sei hier aufgeführt der Befehl, um den Analogeingang 3 auszulesen (zu "pollen"):

<SysEx Header> U3R <EOX>

Die Antwort kommt in einem ähnlichen Format, etwa:

<SysEx Header> U3:237 <EOX>

Als weiteres **Beispiel** der Befehl, um digital I/O Nummer 8 als Ausgang mit High-Pegel zu schalten:

<SysEx Header> K8H <EOX>

Kurzreferenz aller ASCII-Befehle (detaillierte Beschreibungen anschliessend)

UnR	Analogeingang Nr. n auslesen Spezialfälle: UBR alle Analogeing. lesen, UDR digitale Pegel aller Analogeingänge lesen
UD%	Digitalpegel aller Analogeingänge als Folge von 0 und 1 auslesen Reihenfolge der Ausgabe: Analog In Nr.7 zuerst, Nr.0 zuletzt
KnZ	Digital I/O Nr.n als hochohmigen Eingang konfigurieren (n=0 bis 9,A,B)
KnP	Digital I/O Nr.n als Eingang m. Pull-Up Widerstand konfig. (n=0 bis 9,A,B)
KnL	Digital I/O Nr.n als Ausgang LOW konfigurieren (n=0 bis 9,A,B)
KnH	Digital I/O Nr.n als Ausgang HIGH konfigurieren (n=0 bis 9,A,B)
KnR	Digital I/O entprellt als 0 oder 1 auslesen (n=0 bis 9,A,B)
KnQ	Digital I/O unmittelbar, nicht entprellt als 0 oder 1 auslesen (n=0 bis 9,A,B)
KDR	Alle Digital I/O kollektiv entprellt als Zahl auslesen. In die Zahl geht I/O Nr. n mit dem Gewicht 2^n ein.
KDQ	Alle Digital I/O kollektiv unmittelbar als Zahl auslesen
KD%	Alle Digital I/O kollektiv entprellt als Folge von 0 und 1 auslesen Reihenfolge der Ausgabe: I/O Nr.B zuerst, I/O Nr.0 zuletzt
Yn	Zeitkonstante der I/O Entprellung einstellen (ca. n Millisekunden)
SZx	Alle Digital I/O kollektiv als hochohmigen Eingang konfigurieren Parameter ist eine Zahl x, in die jeder zu modifizierende I/O Nr. n mit dem Gewicht 2^n eingeht.
SPx	Alle Digital I/O kollektiv als Eingang mit Pull-Up Widerstand konfigurieren
SLx	Alle Digital I/O kollektiv als Ausgang LOW konfigurieren
SHx	Alle Digital I/O kollektiv als Ausgang HIGH konfigurieren
W0Hn, W2Hn, W6Hn	PWM0, PWM2 bzw. PWM6 mit High-Phase n starten
GHn	Frequenzgenerator High Phase auf n einstellen
GTn	Frequenzgenerator Gesamt-Periode auf n einstellen und starten
Nt	Automatische Nachricht mit Schwelle t konfigurieren bei Analogeingang: UnNt : t=0 bis 255, bei digital I/O: KnNt : t= 0,1
J1R, J3R, J7R, J9R	Aufwärts-Zähler auslesen
X1R, X3R, X7R, X9R	Encoder-Zähler auslesen
J1Z, J3Z, J7Z, J9Z	Aufwärts-Zähler und Encoder-Zähler zurücksetzen
on	diverse Einstellungen zur Konfiguration automatischer Nachrichten (o formal klein geschrieben zur besseren Unterscheidung von Null. Wird intern in Grossbuchstaben umgewandelt)

\$	ab jetzt Parameter in HEX eingeben und auslesen
&	ab jetzt Parameter DEZIMAL eingeben und auslesen
~	I/O und Konfiguration als Preset sichern
@	Preset in I/O und Konfiguration laden
	I/O und Konfiguration auf Lieferzustand zurückstellen, Preset bleibt unverändert

Liste und Beschreibung aller SysEx / ASCII-Befehle

Jeder Steuerbefehl und jede Zustandsmeldung beginnt mit einem charakteristischen Buchstaben. Wenn ein Befehl einen Parameter benötigt, wird er nachfolgend in eckigen Klammern <...> aufgeführt. Diese Klammern dienen nur zur Kennzeichnung von Parametern innerhalb des Manuals, sie sind nicht Teil der zu übertragenden Befehlssequenzen.

Die Befehle für die analogen Eingänge und digitalen I/O setzen sich aus bis zu vier Komponenten zusammen: I/O-Methode ("Device"), I/O-Nummer und Aktionsbefehl mit Parameter.

U <Nummer>

Analog-Eingang für nachfolgende Aktion auswählen.

Parameter: Eingangs-Nummer (Bereich 0 bis 7)

Beispiel: U3 lässt nachfolgende Aktionsbefehle auf den Analogeingang Nr.3 wirken, z.B. **U3R** veranlasst das sofortige Auslesen der momentanen Eingangsspannung und **U3N20** aktiviert automatische Meldungen sobald sich die Eingangsspannung um 20 Stufen verändert hat.

Sonderfälle: UBR liest alle Analogeingänge nacheinander aus. Datenformat für jeden Eingang wie bei Einzelabfrage.

UDR liest die logischen Pegel aller Analogeingänge als kombiniertes Datenwort. Das Format der Meldungen entspricht der Antwort auf Befehl KDR

UD% liest die logischen Pegel aller Analogeingänge als Folge von ASCII 0 und 1. Das Format der Meldungen entspricht der Antwort auf Befehl KD%.

UDN1 aktiviert automatische Meldungen der digital ausgewerteten Analogeingänge. Das Format entspricht der Antwort auf Befehl UDR.

K <Nummer>

Digitalen ("Kombi") I/O für nachfolgende Aktion auswählen.

Parameter: Eingangs-Nummer (Bereich 0 bis 9, A für I/O Nr.10, B für I/O Nr.11).

Beispiel: K8 lässt nachfolgende Aktionsbefehle auf den digital I/O Nr.8 wirken, z.B:

K8R veranlasst das sofortige Auslesen des momentanen logischen Eingangspegels.

K8N0 de-aktiviert automatische Meldungen an K8.

KAL konfiguriert KA (=10) als Ausgang mit Low-Pegel.

Sonderfall: KDR und **KDQ** lesen die logischen Zustände **aller** I/O in einem Datenwort aus. Jedes Bit dieses Datenworts ist gleich dem logischen Pegel der entsprechenden I/O - Nummer.

Beispiel: Wenn alle digitalen I/O als Eingänge mit Pull-Up-Widerstand konfiguriert sind (Lieferzustand ohne extern angeschlossene Komponenten), dann liefert der Befehl **KDR** das Ergebnis:"KD:4095" oder bei hexadezimaler Zahlenbasis: "KD:\$FFF".

Sonderfall: KD% liest die logischen Zustände aller I/O als Folge von 0 und 1 aus. Reihenfolge der Ausgabe: I/O Nr.B zuerst, I/O Nr.0 zuletzt.

Wenn ein Pulsweitemodulator oder der Frequenzgenerator aktiv ist, wird das entsprechende Bit als "g" ausgegeben

L (kein Parameter)

Konfiguriert zuvor ausgewählte digitale I/O als Ausgang LOW

Beispiele: K7L, SL32

H (kein Parameter)

Konfiguriert zuvor ausgewählte digitale I/O als Ausgang HIGH

Beispiele: K7H, \$SHF07

Z (kein Parameter)

Konfiguriert zuvor ausgewählte digitale I/O als hochohmigen Eingang ("highZ")

Beispiele: K7Z, SZ1

Alternative Bedeutung: Impulzzähler und Encoder-Zähler zurücksetzen: J3Z, J9Z

P (kein Parameter)

Konfiguriert zuvor ausgewählte digitale I/O als Eingang mit Pull-Up Widerstand

Beispiele: K7P, SP4095

SL, SH, SZ, SP <Maske>

Kollektiver Setz-Befehl für die digitalen I/O.

SL setzt ausgewählte I/O kollektiv auf "**Ausgang LOW**"

SH setzt ausgewählte I/O kollektiv auf "**Ausgang HIGH**"

SZ setzt ausgewählte I/O kollektiv auf "**hochohmigen Eingang**"

SP setzt ausgewählte I/O kollektiv auf "**Eingang mit Pull-Up Widerstand**"

Parameter: Maske

Um eine differenzierte Modifikation aller I/O Leitungen zu ermöglichen, **wirkt dieser Befehl immer und nur auf diejenigen I/O Leitungen, deren entsprechendes Bit gesetzt (=1) ist.** Die Leitungen, deren entsprechendes Bit nicht gesetzt ist (=0), werden nicht beeinflusst.

Bei der praktischen Umsetzung werden die Bits der zu manipulierenden I/O folgendermassen **zu einer Zahl** kombiniert: **aus dem zu manipulierenden I/O Nr. n wird der Exponent 2^n gebildet und die daraus entstehenden Teil-Zahlen zur Gesamt-"Maske" addiert.**

Beispiel: I/O Nr. 0, 1, 3 und A(=Nr.10) sollen als Ausgang Low gesetzt werden, alle anderen sollen nicht verändert werden: $2^0 = 1$, $2^1 = 2$, $2^3 = 8$, $2^{10} = 1024$, alles addiert = 1035

... also lautet der Befehl SL1035

R (kein Parameter)

"READ" = Befehl für Auslesen ("pollen") eines Eingangswerts

Anmerkung: Die Antwort kommt in folgendem Format:

- Wiederholung des Codes für die abgefragte I/O Technik ("Device")
- ggf. I/O Nummer
- Doppelpunkt
- Ergebniswert
- <carriage return>
- ">"

Beispiele: **U3R** antwortet z.B. **U3:\$3FF** <carriage return>
KAR antwortet z.B. **KA:0** <carriage return>
KDR antwortet z.B. **KD:\$780** <carriage return>
M9R antwortet z.B. **M9:344** <carriage return>

N <Schwelle>

Automatische Nachrichten für einen analogen Eingang oder digitalen I/O einstellen Anwendbar auf die Befehle ("Devices") U und K

Das Format aller automatischen Nachrichten ist gleich dem des entsprechenden R-Befehls.

Parameter: Schwelle der Eingangs-Änderung für neue Nachricht (0 bis 255)
Mit **Schwelle 0** wird die betreffende automatische Nachricht **AUSgeschaltet**.

Bei analogen Eingängen bedeutet **Schwelle 1**, dass **JEDE** Änderung des Eingangssignals gemeldet wird. Wenn die tatsächliche Eingangsspannung zwischen 2 diskreten Stufen des A/D-Wandlers liegt, kann technisch bedingt eine nicht aussagekräftige permanente Serie von Meldungen ausgelöst werden. Diese Vorgabe sollte also mit Bedacht gewählt werden.

Automatische Nachrichten von digitalen I/O und digital ausgewerteten Analogeingängen (UDN1) können nur mit dem Parameterwert 1 eingeschaltet werden.

Beispiele: **U5N8** veranlasst ab sofort automatische Meldungen des **analogen Eingangs Nr.5** sobald sich die im relativen binären Maßstab ausgewertete Eingangsspannung um mindestens 8 gegenüber der vorherigen Nachricht verändert hat. **U5N0** schaltet diese Nachrichten wieder aus.

K5N1 schaltet dagegen automatische Nachrichten des **digital I/O Nr. 5** ein. **K5N0** schaltet diese Nachrichten wieder aus.

Sonderfall: Mit dem Befehl "o1" kann eingestellt werden, dass bei Änderung **eines beliebigen aktivierten I/O Pegels** eine automatische Nachricht **aller digitalen I/O-Pegel** (entprellt) ausgegeben wird. Datenformat siehe KDR-Befehl. Der Befehl "o0" schaltet um auf Ausgabe einzelner I/O Pegel.

Y <Zeitkonstante>

Stellt die Zeitkonstante zur Entprellung der digitalen I/O ein (ca. in Millisekunden)

Parameter: Zeitverzögerung zwischen 2 Abtastungen und Vergleichen der logischen Pegel der digitalen I/O. in Millisekunden

Der permanente Vergleich der Eingangspegel mit dem vorherigen Sample wird automatisch als Hintergrundprozess durchgeführt. Je nach Anwendung ist es sinnvoll, diesen Vergleich in sehr kurzen Zeitintervallen oder mit einer gewissen zeitlichen Trägheit durchzuführen. Zum Entprellen von Relaiskontakten ist häufig 10 Millisekunden ein passender Wert, beim Entprellen einfacher Tastschalter wählt man eher 40-100 Millisekunden.

Die Voreinstellung ist 40 Millisekunden-

Da der Abtastprozess nicht mit den externen Signalveränderungen synchronisiert ist, dauert es mindestens eine und bis zu 2 Abtast-Zeitkonstanten, um eine entprellte Eingangsveränderung festzustellen.

Beispiel: Y10

o <Format>

Konfiguriert das Format automatischer Nachrichten

(o formal klein geschrieben zur besseren Unterscheidung von Null. Wird intern in Grossbuchstaben umgewandelt)

Parameter: Format wie im entsprechenden PROGRAM CHANGE Befehl:

Format = 0: Automatische Nachrichten der digitalen I/O werden individuell pro I/O gesendet (default). (Beispiel: K3:0)

Format = 1: Automatische Nachrichten der digitalen I/O werden simultan für alle I/O gesendet

Format = 16 (hex10): Automatische Nachrichten als Kanalnachrichten senden (default).

Format = 17 (hex11): Automatische Nachrichten im SysEX Format senden.

GT <Gesamt-Periodendauer>

Digital I/O Nr. 0 als Frequenzgenerator aktivieren.

Parameter: Gesamt-Periodendauer des Ausgangspuls als Vielfaches von ca. 21 Mikrosekunden (exakt 21,333us), die maximal einstellbare Frequenz ist ca. 23 Kilohertz (W=1,G=2). Zulässige Parameterwerte sind 1 bis 65535 (hexFFFF). Mit dem maximalen Parameter wird die **minimale Frequenz**, ca. 0,7 Hz eingestellt.

Anmerkung: Der Frequenzgenerator wird mit 2 Befehlen aktiviert: **Bevor der Frequenzgenerator gestartet werden kann, muss zuvor die Dauer der High-Phase** mit dem Befehl GH eingetragen werden. Andernfalls ist mit undefiniertem Verhalten zu rechnen. Bei aktivem Frequenzgenerator können Gesamtperiode und High-Dauer unabhängig voneinander mit sofortiger Wirkung modifiziert werden. **Die Gesamtperiode muss stets größer als die High-Dauer sein!**

Jeder Befehl zum Einstellen des digital I/O Nr. 0 oder zum Einschalten des Pulsweitenedulators W0 schaltet den Frequenzgenerator AUS.

Beispiele: **GH100T200** erzeugt eine Rechteckschwingung von ca. 234 Hz mit 50% Tastverhältnis.
\$GH14TFFF erzeugt positive Nadelimpulse von 426 Mikrosekunden Dauer bei ca. 87 ms Gesamtperiode (=ca. 11Hz).

GH <High-Phase>

Voreinstellung der High-Phase für I/O Nr. 0 als Frequenzgenerator.

Parameter: Dauer der High-Phase des Ausgangspuls als Vielfaches von ca. 21 Mikrosekunden. Zulässige Parameterwerte im Bereich 1 bis 65534 (hex FFFE)

W0H <High-Phase>

Digital I/O Nr. 0 als Pulsweitenedulator aktivieren. In seiner Standard-Anwendung arbeitet der Pulsweitenedulator mit einer konstanten Frequenz von 5865 Hz und kann mit 10 Bit Auflösung (d.h. in 1023 = hex3FF Schritten) justiert werden.

Parameter: High-Anteil des Pulsverlaufs an I/O Nr.0.

In Prozent ausgedrückt: $100 \times \text{<High-Phase>} / 1023$

Anmerkung: Komplexere Einstellmöglichkeiten sind in Anhang A beschrieben
Jeder Befehl zum Einstellen des digital I/O Nr. 0 oder zum Einschalten des Frequenzgenerators schaltet den Pulsweite-Modulator AUS.

Beispiel: W0H256 erzeugt an I/O Nr. 0 eine Rechteckschwingung mit 25% High-Anteil

W2H <High-Phase>

Digital I/O Nr. 2 als Pulsweitenedulator aktivieren. In seiner Standard-Anwendung arbeitet der Pulsweitenedulator mit einer konstanten Frequenz von 5865 Hz und kann mit 10 Bit Auflösung (d.h. in 1023 = hex3FF Schritten) justiert werden.

Parameter: High-Anteil des Pulsverlaufs an I/O Nr.2.

In Prozent ausgedrückt: $100 \times \langle \text{High-Phase} \rangle / 1023$

Anmerkung: Komplexere Einstellmöglichkeiten sind in Anhang A beschrieben
Jeder Befehl zum Einstellen des digital I/O Nr. 2 schaltet den Pulsweite-Modulator AUS.
Solange W2 aktiv ist, kann Encoder X1 nicht ausgewertet werden.

W6H <High-Phase>

Digital I/O Nr. 6 als Pulsweitemodulator aktivieren. In seiner Standard-Anwendung arbeitet der Pulsweitemodulator mit einer konstanten Frequenz von 23529 Hz und kann mit 8 Bit Auflösung (d.h. in 255 = hexFF Schritten) justiert werden.

Parameter: High-Anteil des Pulsverlaufs an I/O Nr.6.

In Prozent ausgedrückt: $100 \times \langle \text{High-Phase} \rangle / 255$

Anmerkung: Komplexere Einstellmöglichkeiten sind in Anhang A beschrieben
Jeder Befehl zum Einstellen des digital I/O Nr. 6 oder zum Einschalten des Frequenzgenerators schaltet den Pulsweite-Modulator AUS.

Beispiel: W6H128 erzeugt an I/O Nr. 6 eine Rechteckschwingung mit 50% High-Anteil

J1R , J3R , J7R , J9R (kein Parameter)

Impuls-Zähler auslesen.

Der Zähler wird bei jeder **negativen Flanke der I/O Leitung** um eins heraufgezählt. Dabei spielt es keine Rolle, wie die Zählflanke erzeugt wird: durch ein externes Steuersignal oder durch einen entsprechenden Ausgangsimpuls. Der Zähler unterscheidet bereits Eingangsimpulse, die nur wenige Mikrosekunden nacheinander eintreffen. Daher ist bei vielen externen Impulsgebern, wie Tastschaltern, eine geeignete Entprellung durch externe Schaltungsmassnahmen notwendig, z.B. mit Tiefpassfiltern.

Anmerkung: Impulszähler und Encoder-Zähler verwenden getrennte Speicherzellen, d.h. können unabhängig voneinander ausgewertet werden. Der Rücksetz-Befehl wirkt jedoch stets auf beide gemeinsam.

Beispiel: J3R liest den aktuellen Stand des Impulszählers an I/O Nr. 3 aus

J1Z , J3Z , J7Z , J9Z (kein Parameter)

Impuls-Zähler und Encoder-Zähler auf 0 zurücksetzen.

Anmerkung: Der Rücksetz-Befehl wirkt stets auf beide gemeinsam.

Beispiel: J3Z setzt Aufwärts-Zähler J3 und Encoder-Zähler X3 gemeinsam zurück

X1R , X3R , X7R , X9R (kein Parameter)

Encoder-Zähler auslesen.

Jeder Encoder-Zähler zählt **aufwärts**, wenn während der negativen Zählflanke der Eingang mit der nächsthöheren Nummer (2, 4, 8 oder A) auf **High-Pegel** liegt, zählt jedoch **abwärts**, wenn der Nachbar I/O auf **Low-Pegel** liegt. Dabei spielt es keine Rolle, wie der Pegel eingestellt wird: durch ein externes Steuersignal oder durch entsprechende Schaltung als Ausgang.

Beispiel: M9R liest den Stand des Encoder-Zählers an I/O Nr. 9 mit Richtungseingang A aus

Systembefehle:

\$ (kein Parameter)

Setzt die Zahlenbasis auf Hexadezimal.

Anmerkung: Alle Parameterwerte der Befehle werden von nun an als Hexadezimalzahlen interpretiert bzw. in Meldungen als solche ausgegeben. Dieser Zustand bleibt aktiv, bis durch einen entsprechenden Befehl eine andere Zahlenbasis eingestellt wird. Da die Zahlenbasis auch im Preset gespeichert ist, wird evtl. auch durch das Laden des Presets oder durch ein Reset der Sensorbox die Zahlenbasis verändert.

& (kein Parameter)

Setzt die Zahlenbasis auf Dezimal.

Anmerkung: Die Parameterwerte aller Befehle werden von nun an als Dezimalzahlen interpretiert bzw. in Meldungen als solche ausgegeben. Dieser Zustand bleibt aktiv, bis durch einen entsprechenden Befehl eine andere Zahlenbasis eingestellt wird. Da die Zahlenbasis auch im Preset gespeichert ist, wird evtl. auch durch das Laden des Presets oder durch ein Reset der Sensorbox die Zahlenbasis verändert.

~ (kein Parameter)

Speichert die aktuelle I/O Konfiguration permanent

Beim Einschalten oder Reset wird grundsätzlich automatisch das Preset geladen und aktiviert.

@ (kein Parameter)

Lädt das Preset und stellt die gespeicherte I/O Konfiguration wieder ein

| (kein Parameter)

Alle I/O Leitungen und sonstige Konfigurations-Einstellungen werden auf den Grund-Zustand gesetzt:

- Zahlenbasis "Dezimal"
- Alle digitalen I/O werden als Eingang mit Pull-Up Widerstand konfiguriert.
- Alle automatischen Nachrichten werden ausgeschaltet

Das Preset und die Geräte-ID bleiben unverändert.

Anhang A:

Komplexere Konfiguration des Pulsweite-Modulators an I/O Nr. 0 und 2:

Mit der im Hauptteil beschriebenen Konfiguration kann der Pulsweitemodulator gut eingesetzt werden zur **Erzeugung analoger Ausgangsspannungen** mit Hilfe eines externen Tiefpassfilters. Mit den nachfolgend beschriebenen Alternativen eröffnen sich andere Anwendungen, z.B. Ansteuerung von Modellbau-Servos.

Pulsweitemodulator an I/O Nr.2 kann nur aktiviert werden, während I/O Nr. 0 ebenfalls als Pulsweitemodulator arbeitet. Die Grundfrequenz und die Auflösung von W2 wird immer bestimmt durch die Einstellung von W0

Hi Arg. (hex)	Hi Arg. (dez)	Auflösung (Bit)	Basisfrequenz (Hz)	Low Arg.(hex)	Low Arg. (dez)
1000	4096	8	23529	0 .. FF	0..255
2000	8192	8	2941	0 .. FF	0..255
3000	12288	8	368	0 .. FF	0..255
4000	16384	8	92	0 .. FF	0..255
5000	20480	8	23	0 .. FF	0..255
9000	36864	10	5865	0 .. 3FF	0..1023
A000	40960	10	733	0 .. 3FF	0..1023
B000	45056	10	92	0 .. 3FF	0..1023
C000	49152	10	23	0 .. 3FF	0..1023
D000	53248	10	5.7	0 .. 3FF	0..1023

Das Konzept ist am besten verständlich, wenn man es in Hexadezimal-Darstellung betrachtet. **Das oberste Nibble ("Hi-Arg")** beschreibt die **Basisfrequenz und Auflösung, die unteren 3 Nibbles definieren den High-Anteil der erzeugten Rechteck-Schwingung**. Bei dezimaler Denkweise muss man einfach beide Anteile als "krumme" Zahlen addieren.

Übrigens: die im Hauptteil beschriebene Konfiguration ist identisch mit der hier unter Hi Arg hex9000 beschriebenen Konfiguration. Beim Neustart muss immer das Hi-Arg explizit angegeben werden (abgesehen von o.g. Ausnahme), **soll jedoch bei aktiver PWM das Low-Argument verändert werden, reicht es aus, das High-Argument gleich 0 zu setzen.**

Komplexere Konfiguration des Pulsweite-Modulators an I/O Nr. 6:

Dieser Pulsweitemodulator erlaubt nur 8 Bit Auflösung.

Hi Arg. (hex)	Hi Arg. (dez)	Basisfrequenz (Hz)	Low Arg.(hex)	Low Arg. (dez)
0 oder 100	256	23529	0 .. FF	0..255
200	512	11765	0 .. FF	0..255
300	768	2941	0 .. FF	0..255
400	1024	735	0 .. FF	0..255
500	1280	368	0 .. FF	0..255
600	1536	92	0 .. FF	0..255
700	1792	23	0 .. FF	0..255

Anhang B: Einsatz als Standalone-Steuergerät

Aufgrund der Komplexität des Themas können hier nur die Haupteigenschaften angedeutet werden. Die **komplette Dokumentation** des in der Firmware enthaltenen alternativen programmierbaren Kommandointerpreters finden Sie auf der Webseite "www.as-control.de" oder auf der mitgelieferten CD-ROM.

Der programmierbare Kommandointerpreter wird in den Codierschalter-Stellungen 0 , E und F aktiviert anstelle des in diesem Manual beschriebenen Standard-Interpreters.

Da es keine brauchbaren Terminalprogramme mit MIDI-Baudrate gibt, startet die MIDI-Sensorbox in **Schalterstellung E mit 19200 Baud und in Stellung F mit 115200 Baud**. Auf unserer Webseite ("www.cinetix.de/interface/tiptrix/") ist die **Hardware-Konfiguration eines Verbindungskabels von MIDI auf serielle Schnittstelle** beschrieben. So kann die Programmierung mit einem üblichen ASCII-Terminalprogramm vorgenommen werden.

In Stellung 0 des Codierschalters startet die MIDI-Sensorbox dagegen mit MIDI-Baudrate, hat aber die gleiche textorientierte Steuersyntax.

In allen 3 Schalterstellungen werden SysEx Header ohne Zutun des Anwenders herausgefiltert, so dass es für ihn egal ist, ob er Befehle in "plain ASCII" schickt oder diese in SysEx Pakete packt. Das Echo auf SysEx Befehlszeilen kommt stets als SysEx Meldung zurück, auf reine ASCII-Befehlszeilen kommt auch eine reine ASCII-Rückmeldung. Die Programmierung ausschliesslich mit SysEx Befehlen ist prinzipiell möglich, aber unkomfortabler als mit reinem ASCII Text.

Zusätzlich ist auf niedriger Systemebene ein Filter für MIDI-Kanalnachrichten eingebaut. Beliebig in oben beschriebene Kommandozeilen eingeflochtene oder einzeln gesendete MIDI-Kanalnachrichten werden herausgefiltert, in einem speziellen Puffer abgelegt und können asynchron vom Standalone-Programm ausgewertet werden. Zur klaren Trennung von Kanalnachrichten gegenüber ASCII Text wird unbedingt empfohlen, **keine Kanalnachrichten mit "Running State" zu senden!** So ist es möglich, den Programmablauf z.B. mit einem MIDI-Keyboard zu lenken oder solchen Input mit der Auswertung der I/O Leitungen zu verknüpfen. Hierzu sind die folgenden Spezial-Befehle implementiert: MCHECK, GETMSG, DROPMSG, STATUS, MDAT1, MDAT2.

MCHECK entspricht dem "kbhit" der C-Programmierung. Statt eines TRUE Flag gibt es das Statusbyte der nächsten im Puffer anstehenden (= ältesten dort vorhandenen, Warteschlangenprinzip) MIDI-Nachricht zurück, lässt die Nachricht aber im Puffer.

GETMSG holt die nächste Kanalnachricht aus dem Puffer (oder wartet darauf wenn noch keine da ist) und speichert ihre Bestandteile ab in den Systemvariablen

STATUS, MDAT1 und MDAT2 – die dann anschliessend mit den gleichnamigen Befehlen ausgelesen werden können. Sie werden erst mit dem nächsten GETMSG Befehl überschrieben.

DROPMSG entfernt die am längsten anstehende Kanalnachricht aus dem Puffer ohne sie in die o.g. Systemvariablen zu kopieren.

Umgekehrt können natürlich auch z.B. aus den Spannungseingängen **beliebige MIDI Nachrichten** wie etwa NOTE ON oder CONTROL CHANGE Befehle hergeleitet und **über MIDI OUT gesendet werden**.

Abgesehen von diesen - für ein Gerät mit MIDI-Schnittstelle notwendigen Spezialfunktionen - erfolgt die Programmierung mittels einer Syntax, die einem sehr vereinfachten "C" bzw. "Java" entspricht. Hierbei handelt es sich um einen schnellen komplett in Assembler programmierten Kommandointerpreter. Anders als bei ähnlichen Projekten erfolgt die Kompilation von Anwenderprogrammen nicht mit einem Spezialprogramm vorab auf dem PC, sondern direkt im AS-Control Interpreter.

Vereinfachungen und Modifikationen der C-Syntax wurden vor allem vorgenommen um Speicher zu sparen und um ein möglichst effizientes experimentelles interaktives Arbeiten zu ermöglichen. Z.B. gibt es keine Funktions-Prototypen und kein "main" Programm. Jede

Funktion kann jederzeit für sich vom Terminal oder per SysEx Nachricht aufgerufen werden (ob das jeweils Sinn macht entscheiden Sie als ProgrammiererIn).



Cinetix Medien und Interface GmbH
Gemündenerstr. 27 D-60599 Frankfurt/Main
Fon: +49-69-68 51 05 Fax +49-69-68 600 409
<http://www.cinetix.de/interface/>

* Technische Änderungen, Irrtum und Lieferbarkeit vorbehalten.

* Diese Beschreibung ist informativ und sichert keine Produkteigenschaften im rechtlichen Sinne zu.

* Im Text zitierte Warenzeichen und Produktnamen sind geschütztes Eigentum ihrer Eigentümer