

Carsten Meyer

Spielhilfe

Der c't-Klangcomputer II: Musik-Workstation zum Selbstbau

Virtuelle Instrumente erobern die Musikstudios: Was früher mit unzähligen Oszillatoren, Filtern und Hüllkurvengeneratoren in aufwendige Hardware gegossen wurde, lässt sich dank der enormen Rechenleistung aktueller Prozessoren in Software perfekt emulieren – und mannshohe Synthesizer-Burgen schrumpfen auf ein paar Megabyte im PC-Hauptspeicher. Mit etwas selbst gebauter Hardware und ein paar Kniffen lässt sich der PC auch als Live-Instrument auf der Bühne nutzen.



Dem Bühneneinsatz virtueller Instrumente steht vor allem das unpraktische Handling eines üblichen PC-Systems im Wege. Ein Notebook ist hier leider nur ein empfindlicher, selten Roadie-fester und noch dazu teurer Ersatz. Unser Ansatz basiert auf einem preiswerten PC-Mainboard, das mit Peripherie, Klaviatur und Bedienelementen in ein gemeinsames Gehäuse verfrachtet wird und so ein Standalone-Gerät ergibt, das den rauen Bühnenalltag nicht übel nimmt und dabei nicht mehr wiegt als ein digitales Stage-Piano.

Das kommt Ihnen bekannt vor? c't-Leser der ersten Tage erinnern sich vielleicht noch an ein Projekt, das wir vor 20 Jahren vorstellten: den c't-Klangcomputer. Das Gerät wartete mit seinerzeit richtungsweisenden Features wie Sampling, Fourier- und FM-Synthese auf, wie sie sonst nur um Zehnerpotenzen teurere Studio-Systeme boten. Heute mutet die damals verwendete Hardware wahrlich anachronistisch an: Ein Apple-II-Board mit sagenhaften 48 KByte RAM (man beachte das K vor „Byte“) wurde durch Steckkarten zum mehrstimmigen Wave-table-/Sampling-Synthesizer erweitert und erschloss dem Bast-

ler (fast) dieselben Möglichkeiten wie der sündhaft teure PPG-Wave-Synthesizer oder das Fairlight-CMI-System. Ein Großteil der Klangerzeugung wurde noch mit handfester analoger Schaltungstechnik realisiert, denn für eine Echtzeit-Synthese war der ein einziges MHz schnelle 6502-Prozessor nicht zu gebrauchen. Sehr wohl konnte er aber die komplette Steuerung, das Sampling und die Klangberechnung übernehmen, auch wenn das Erstellen eines 8-Bit-FM-Samples Minuten in Anspruch nahm.

Lizenz zum Tröten

In der Zwischenzeit sind Rechenleistung und Speicherkapazität um mindestens den Faktor 3000 angewachsen, und die mit allerlei Signalverarbeitungsbefehlen à la MMX erweiterten Prozessoren neuzeitlicher PCs können ohne weiteres ein Dutzend spezieller Signalprozessoren (DSPs) substituieren. So setzt man in der Desktop-Musikproduktion immer mehr auf Software-Synthesizer, die in Echtzeit die einst zentnerschwere Musikelektronik simulieren. Die Palette an virtuellen Instrumenten reicht von hochwertigen Samplern, Synthesizern mit den

aberwitzigsten Synthese-Algorithmen und Drumcomputern über Alleinunterhalter-Begleitautomaten bis hin zu naturgetreuen Nachbildungen von Gitarrenverstärkern, Effektgeräten und historischen Instrumenten mit all ihren klanglichen Eigenheiten. Kurz gesagt: Das Angebot an virtuellen Klangerzeugern ist unüberschaubar vielfältig und inzwischen mindestens genauso groß wie das von realen elektronischen Musikinstrumenten.

Die meisten dieser Programme können für sich allein (Standalone) laufen oder als Plug-in in ein virtuelles „Rack“ geschoben werden, etwa das eines Sequenzer- oder Recording-Programms. Obwohl etwas umständlicher einzurichten, ist die Plug-in-Lösung die universellere, denn nur hier lassen sich die Audio- und MIDI-Informationen bei mehreren gleichzeitig laufenden Instrumenten und Effekten wie gewünscht aufteilen. Wenn etwa der Klang noch durch ein Software-Hallgerät geleitet und mit eingeschleift, natürlich ebenfalls virtuellen Effekten digital abgemischt werden soll, ist ein Plug-in-Rack Pflicht.

Als Standard für Musiksoftware-Plug-ins hat sich Steinbergs Virtual

Studio Technology (VST) etabliert, die 1996 mit der Musikproduktionssoftware Cubase eingeführt wurde. VST ist ein offener Standard, der die Integration virtueller Effekt-Prozessoren und Instrumente in eine digitale Audiumgebung ermöglicht. Eigentlich wurde VST als Effekt-Software-Schnittstelle für Sequenzer- und Recording-Programme konzipiert, aber es finden sich auch reine Plug-in-Racks mit sehr weit reichenden Patch- und Konfigurationsmöglichkeiten (etwa Steinbergs VStack), inzwischen wie einige VST-Instrumente und -Effekte sogar als Free- und Shareware.

Der errechnete Klang materialisiert sich schließlich im D/A-Wandler einer PCI-Audiokarte. Hierbei stellen die Musik-Programme ganz andere Anforderungen als beispielsweise Spiele – deshalb an dieser Stelle auch die explizite Abgrenzung zur „Soundkarte“. Die so genannte Latenzzeit ist das wichtigste Kriterium. Soundkarten und On-Board-Codex mit Latenzzeiten von 30 ms und mehr sind für den musikalischen Einsatz unbrauch-

bar: Das Spielgefühl leidet bereits bei einer Verzögerung von mehr als 10 ms.

Ebenso wichtig ist eine direkte ASIO-Unterstützung seitens der AUDIO-Hardware. Das herstellerübergreifend akzeptierte Protokoll, ebenfalls eine Steinberg-Entwicklung, ist auf besonders geringe Latenzen ausgelegt und arbeitet bereits mit Buffer-Größen von wenigen hundert Bytes zufriedenstellend, während übliche Soundkarten-Treiber einige Kilobytes an Klangpuffer benötigen, um nicht ins Stottern zu geraten. Brauchbare Audio-Karten mit ASIO-Unterstützung gibt es ab circa 80 Euro. Zum Ausprobieren reicht zwar ein Soundkarten-ASIO-Emulator aus dem Freeware-Fundus, brauchbare Ergebnisse erzielt man damit aber nur in Ausnahmefällen.

Um kurz auf die Kosten zurückzukommen: Die virtuellen Klangerzeuger und Effektgeräte (im VST-Duktus gelten auch Instrumente als Effekte) sind zwar billiger als echte, die leistungsfähigeren kommen trotzdem schon mal auf einige hundert Euro. Als kleines Bonbon hat die auf virtuelle Instrumente spezialisierte Berliner Firma Native Instruments den c't-Lesern drei (etwas abgespeckte) beliebte Instrumente im „XPress“-Paket zum Sonderpreis versprochen. Es handelt sich dabei um den FM7, einen Yamaha-DX7-Clone, den Prophet-Emulator Pro53 und die virtuelle Tonrad-Orgel B4, die verblüffend nach Jimmy Smith oder Bärbel Dennerlein klingen kann – es fehlt nur der Geruch von Schmieröl, Holz und staubigen Elektronenröhren. Die XPress-Versionen bieten zwar lediglich (in Grenzen veränderbare) Presets, zum Einstieg in die Welt der virtuellen Instrumente ist das Paket aber eine solide Grundausstattung.

Band in a Box

Wie vor 20 Jahren kommt auch unser „Klangcomputer II“ ohne eine Klaviatur (der Begriff „Tastatur“ wäre im PC-Umfeld zweideutig) und einige analoge Eingabemöglichkeiten nicht aus. Zwar gibt es gewisse Musiker, die ihre Elaborate komplett an der PC-Tastatur komponieren und einspielen, doch für diese Klientel ist unser Projekt nicht gedacht. Es will vielmehr den PC zu einem Live-tauglichen Perfor-

mance-Instrument machen, indem es fast schon vergessene „analoge“ Bedienungsweisen wieder einführt.

Schließlich gehört die Bedienung von Drehknopf und Schieberegler zum allgemeinen Kulturgut. Wer in der wenig intuitiven und unbequemen Up-Down-Button-Sparmentalität einen technischen Fortschritt sieht, hat noch nie ein altes Dampfradio besessen. Lassen Sie einmal, vielleicht beim nächsten Flohmarktbesuch, die schwungradgetriebene Sendereinstellung eines in Gelsenkirchener Barock gehaltenen Röhren-Supers rotieren, dann wissen Sie, was wir meinen.

Das sehen die Schöpfer der virtuellen Klangerzeuger eigentlich auch so: Statt nackter Zahlen repräsentieren anschauliche Bedienpanel-Oberflächen mit Drehreglern, Fadern und „analogen“ Anzeigen die Einstellung. Allerdings scheint uns die Bedienung eines virtuellen Drehknopfs mit der Maus ein Schritt in die falsche Richtung. Genau hier setzt das Klangcomputer-Projekt an: Nicht die eigentliche Klangerzeugung soll Gegenstand der Artikel sein, sondern die eingängige Bedienung der virtuellen Instrumente. Das hält den Bastel- und Programmieraufwand in Grenzen, und trotzdem kommen Lötkolben und Schraubendreher nicht zu kurz.

Was braucht's?

Auch die modernsten Software-Synthesizer steuert man wie vor Jahr und Tag über MIDI an, wobei die digitalen Steuersignale beispielsweise über einen Adapter am Gameport in den Rechner gelangen. Die MIDI-Steuerung umfasst nicht nur die angeschlagenen Noten auf der Klaviatur, sondern auch diverse „Controller“ [2], mit denen man die virtuellen Drehknöpfe des Software-Instruments bedient. So genannte MIDI-Masterkeyboards verfügen denn auch über diverse Eingabemöglichkeiten, oft als Fader oder als Endlos-Inkrementalgeber in Drehknopf-Form implementiert. Natürlich reagiert deren Klaviatur wie eine alte Schreibmaschine oder eben ein Klavier anschlagdynamisch, und bei den ganz teuren Geräten ist sogar ein „Aftertouch“-Sensor eingebaut, der dem Spiel bei gezieltem Tastenachdrücken einen besonderen Ausdruck verleihen kann – etwa

durch eine Filter-Modulation oder ein dann einsetzendes Vibrato.

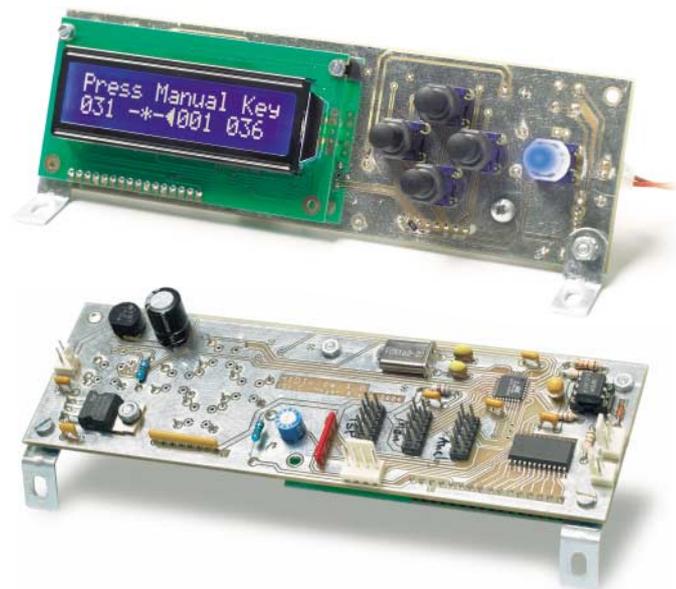
Gut, werden Sie sagen, dann kaufe ich mir halt so ein fertiges Masterkeyboard und steuere damit meinen PC, und irgendwie werden die Sachen schon in ein gemeinsames Gehäuse passen. Stimmt, aber Masterkeyboards mit komfortablen Eingabe- und Steuermöglichkeiten und womöglich einer klavierähnlich gewichteten, großen Tastatur sind teuer. So teuer, dass Sie sich sehr wohl überlegen werden, die 600-Euro-Investition zu zerlegen, um deren Einzelteile für den Einbau in ein auch noch den PC aufnehmendes Gehäuse zu verwerten. Davon abgesehen wollen Sie ja vielleicht auch eigene „analoge“ Spielhilfen wie Blaswandler, Kniehebel, Lichtreflex-Sensoren, Zugriegel, Fußschweller, Drucksensoren und dergleichen anschließen, und dann wird es mit der käuflichen Lösung schwierig. Die vom Hersteller vorgegebene mechanische Ausführung lässt ebenfalls wenig Spielraum, sodass der Eigenbau eines integrierbaren Masterkeyboards die flexibelste und preiswerteste Lösung ist.

Mach flott den Schrott

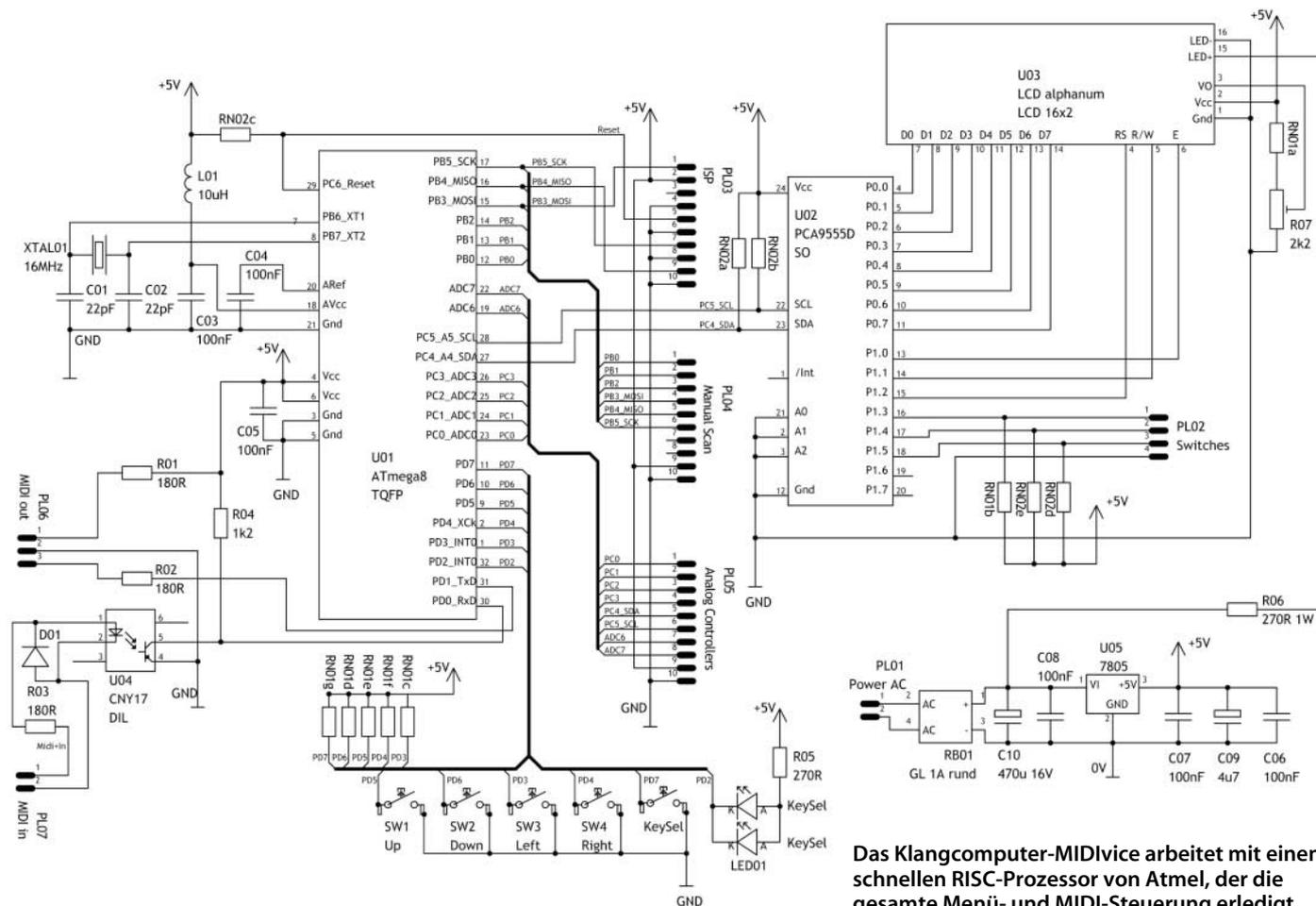
Allerdings wollen wir Sie nicht zu Laubsägearbeiten nötigen, und auch den Kauf einer ferti-

gen Klaviatur können Sie sich sparen. Stattdessen ist Recycling angesagt: Unzählige Dachböden, Kammern und Keller halten Schätze in Form von alten Heimorgeln oder „Tischhupen“ (Stereo-Keyboards) vor, was sich in einem geradezu inflationären Preisgefüge bei Online-Auktionen manifestiert. Eine alte Farfisa Jaqueline oder ein Yamaha-Keyboard der ersten PSR-Generation wechseln oft nicht einmal für einen Euro den Besitzer, selbst einst für viel Geld und mit unendlicher Geduld eigenhändig erbaute Böhm- und Wersiklassiker finden nur noch unterhalb der 50-Euro-Schwelle Käufer.

Auch wenn der Aufbau der eigentlichen Klaviatur und der zugehörigen Scan-Platine erst in der nächsten Ausgabe ansteht, lohnt es doch schon jetzt, nach „Tastbarem“ Ausschau zu halten. Für unser Projekt lässt sich nämlich alles verwerten, was Tasten mit Kontakten drunter hat. Achten Sie aber bei der Klaviatur-Akquise von vornherein auf ein gutes Spielgefühl. Piano-affine Keyboarder legen natürlich Wert auf einen klavierähnlichen Anschlag, hier ist die Auswahl schon deutlich enger. Dennoch sollte ein vordigitales Stage-Piano, das klanglich nicht mehr zu überzeugen weiß, für wenig Geld zu finden sein. Und nicht zuletzt lassen sich auch rein mechanische Tasteninstrumente mit Tastenkon-



Achten Sie bei der Montage des Displays auf einen Sicherheitsabstand von wenigstens einem Millimeter zu den Löt pads der Platine. Displays ohne Hintergrundbeleuchtung haben nur 14 Pins, hier bleiben die Pins 15 und 16 (ganz rechts) offen.



Das Klangcomputer-MIDIvice arbeitet mit einem schnellen RISC-Prozessor von Atmel, der die gesamte Menü- und MIDI-Steuerung erledigt.

takten versehen, wenn man mit einem gesunden Nervenkostüm und bastlerischem Geschick gesegnet ist.

Nicht zuletzt kann die Gehäusefrage die Auswahl der Klaviatur entscheiden. Für ein angenehmes Spielgefühl ist die Länge des Tastenhebels entscheidend, und der fällt bei den

besser spielbaren Tastaturen halt länger aus. Typische Plastik-Klavaturen bauen sehr viel kürzer. Nachteil vieler Heimorgel-Manuale ist der begrenzte Tastenumfang von nur dreieinhalb oder vier Oktaven. Für ein Solo-Instrument mag das ausreichen, beidhändiges Spiel verlangt aber fünf Oktaven und mehr.

Zwar wird es dem gewieften Heimwerker gelingen, zwei kurze Manuale nahtlos aneinander zu reihen, zur allgemeinen Nachahmung ist dies aber weniger zu empfehlen. Unser Tipp: Lassen sie die Heimorgel mechanisch so, wie sie ist, und ersetzen Sie einfach nur die Klang-erzeugung durch den Klang-computer. Eine harmlose Philicorda, die wie Jon Lords C3-Hammond mit 122er-Leslie klinget, hat doch auch was und ist bei Auftritten immer für eine Überraschung gut.

MIDIifikation

Der Hardware-Teil unseres Projekts besteht aus mehreren, zum Teil austauschbaren und optionalen Komponenten, die zusammen ein hochwertiges Master-keyboard mit bis zu 215 Tasten, 16 analogen und 8 digitalen MIDI-Controllern zum Einbau in beliebige Gehäuse ergeben. Wichtigster (und mit rund 50 Euro teuerster) Teil ist das Micro-controller-Board MIDIvice, das mit fünf Bedientastern und einem alphanumerischen LC-Display bestückt ist. Hier können die analogen und digitalen Be-

dienelemente, im Prinzip Potenziometer und Schalter, den für das jeweilige Instrument erforderlichen MIDI-Controllernummern zugeordnet werden. Die Einstellungen und Zuweisungen lassen sich in 16 „Sets“ speichern und auf Knopfdruck abrufen – der Wechsel selbst diffiziler Zuordnungen ist damit kein Problem.

Obligatorisch sind natürlich eine oder mehrere Tastatur-Scan-Platinen, die wir in drei verschiedenen Ausführungen vorstellen werden: Für anschlagdynamische Tastaturen mit Umschaltkontakten, für nicht anschlagdynamische Orgel-Tastaturen mit einfachen Einschaltern und für Tastaturen mit vorhandener, beliebiger Matrix-Anordnung der Tastenkontakte, wie sie in neueren Geräten meist zu finden ist. Die Platinen für Orgel-Tastaturen lassen sich hintereinander schalten, sodass bis zu drei 5-Oktaven-Manuale und ein Vollpedal mit 32 Tasten abgefragt werden können – deshalb die spektakuläre Tastenzahl 215. Für jede Scan-Platinen-Version ist eine angepasste MIDIvice-Firmware erforderlich; dies ist bei der Bestellung zu beachten.

Stückliste	
U01	ATmega8-16AI (TQFP)
U02	PCA9555D (SO)
U03	LCD16x2, Abmessungen 84 mm x 44 mm, z. B. LTN211 Philips
U04	CNY17-II (siehe Text)
U05	7805 Spannungsregler
RB01	B40C800 o. ä., Brückengleichrichter rund
D01	1N4148
LED01	LED 3 mm, Farbe beliebig
C01,C02	22p ker. RM 5 mm
C03...C08	100n ker. RM 5 mm
C09	4µ7/35V Tantal
C10	470µ/25V Elko
R01...R03	180R
R04	1k2
R05	270R
R06	270R 1W (siehe Text)
R07	2k2 Präz.-Trimmer 5 mm stehend
RN01	10k x 7 R-Netzwerk
RN02	10k x 5 R-Netzwerk
L01	10µH Min.-Drossel liegend
XTAL01	Quarz 16 MHz
Sw1...Sw5	Print-Taster D6 rund RM 5 mm oder Deltron-Taster MultiMec 3F mit Tastenkappen, je nach Bauhöhe des Displays
Sonstiges	Platinen-Steckverbinder 2-pol., 3pol., 4-pol., Pfostenstecker 10-pol. zweireihig (3 Stück), Pfostenstecker 16-pol. einreihig (für LCD), Distanzröllchen 5 mm, Schrauben M2,5 x 12 mit Muttern (für LCD), Schraube M3 x 8 mit Mutter für 7805

Als dritte Komponente stellen wir eine I/O-Platine mit 16 analogen und acht Schaltereingängen sowie vier Schaltausgängen vor. Die analogen Eingänge nehmen Steuerspannungen beliebigen Ursprungs entgegen, etwa vom Pitch-Bend-Rad, einem Poti-Bedienpanel oder einem Aftertouch-Sensor. Selbst Drucksensoren, Fußschweller und Reflexlichtschranken lassen sich so mit wenig Aufwand anschließen und „midifizieren“. Die Schaltereingänge lassen sich ebenfalls beliebigen MIDI-Controllernummern zuordnen, wobei hier natürlich nur zwei Zustände – an und aus – übermittelt werden. Besonderer Pfiff des I/O-Boards sind vier Schaltausgänge, die abhängig vom im „Set“ eingestellten Zustand zum Steuern externer Effekte herangezogen werden können. Wer will, kann hier Relais zum Umschalten des analogen Verstärker- und Effektwegs oder gar zum Einschalten von Lichteffekten anschließen. Auch hier finden nur sehr preiswerte Standardbauteile Verwendung, keine der Peripherie-Platinen kommt vom Bauteileaufwand nennenswert über zehn Euro.

Tastenfrage

Wichtigster Baustein des MIDIvice-Boards ist ein schneller RISC-

Mikrocontroller vom Typ Atmel ATmega8. Die Wahl dieses Bausteins fiel angesichts seiner weiten Verfügbarkeit, der einfachen Programmierung und des moderaten Preises von etwa drei Euro leicht. Man benötigt noch nicht einmal ein Programmiergerät, um ihm seine Firmware einzuverleiben – dazu genügt ein mit zwölf Euro sehr preiswerter ISP-Adapter [4] und ein Freeware-Tool (Lancos PonyProg [5]) zum Flashen. Auf der Platine ist ein In-System-Programming-Steckverbinder (ISP) vorgesehen, der mit diesen Tools und den gängigen Atmel-In-Circuit-Programmern (STK200/500) kompatibel ist, was Änderungen oder ein Austausch der Firmware ohne Ausbau des Boards ermöglicht. Die übrigen Portleitungen des ATmega8 führen zum Scan-Anschluss für die Tastatur-Platine(n) und zum Steckverbinder der Analog-I/O-Platine.

Des Weiteren findet sich auf dem Board ein LC-Display in preiswerter Industriestandard-Ausführung, das über einen I2C-Portbaustein an den Mikrocontroller angekoppelt ist. An seine freien Portleitungen werden später nach Bedarf auch Sustain- und Sostenuto-Pedal angeschlossen. Ein kleines Netzteil mit Brückengleichrichter und Span-

nungsregler rundet die Schaltung ab; die Versorgung kann über einen Steckertrafo oder über die +12V-Leitung des PC-Netzteils geschehen. Der eigene Spannungsregler ist erforderlich, um eine stabile Referenzspannung für die internen A/D-Wandler des Mikrocontrollers zu gewährleisten.

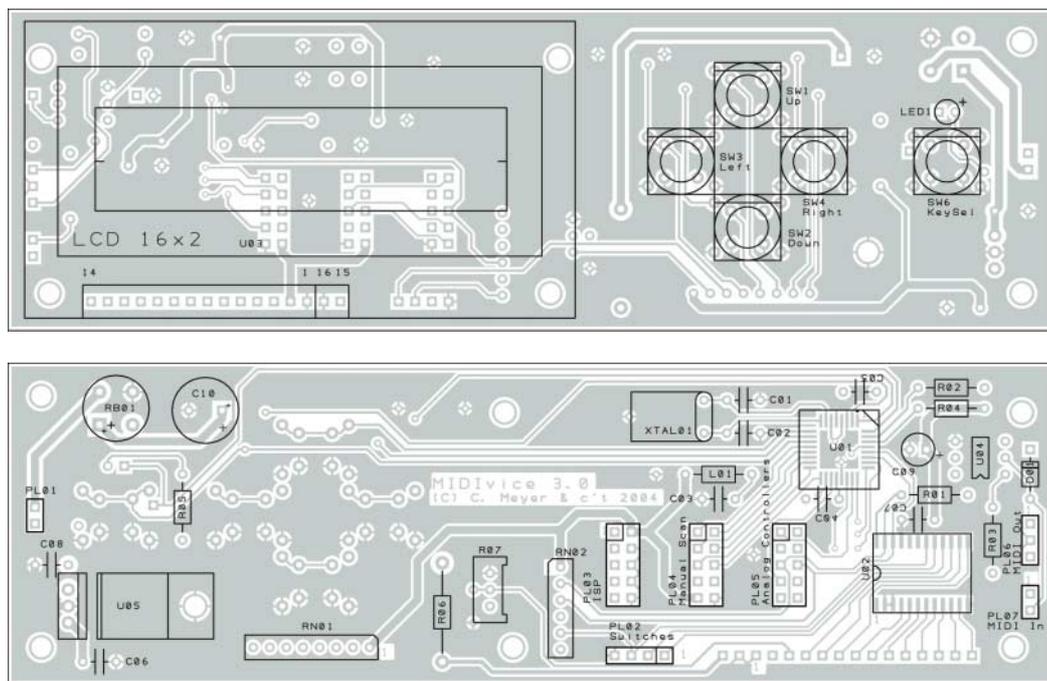
Die Schaltungsbeschreibung des MIDIvice-Boards wäre damit fast abgeschlossen, gäbe es nicht die Firmware-Frage zu diskutieren. Die Aufgabe des Programms ist eigentlich schnell umrissen: Abfrage der Tastatur und der Eingänge, Abfrage der Bedientaster, um eine Einstellung verschiedener Parameter zu ermöglichen, und nicht zuletzt das Senden der richtigen MIDI-Befehle. So unscheinbar die Schaltung zunächst aussieht: Der größte Entwicklungsaufwand steckt in der 8 Kilobyte großen Firmware.

Die Tastatur-Abfrage ist bei anschlagdynamischen Tastaturen keineswegs trivial. Das Programm misst im Prinzip die Geschwindigkeit, mit der ein Umschalter unter jeder Taste von einem zum anderen Kontakt wechselt. Je kleiner diese Zeit, desto heftiger wurde diese Taste gedrückt. Die empirisch ermittelte Umschaltzeit liegt bei einem normalen Tastatur- und damit Kontakthub ungefähr zwischen 0,5 und 30 Millisekun-

den. Um eine ausreichende zeitliche Auflösung zu erzielen – MIDI lässt immerhin 127 verschiedene Dynamik-Stufen zu –, ist der Zustand der Taste also etwa alle 250 Mikrosekunden abzufragen. Da der Controller nebenbei noch andere Aufgaben zu erledigen hat, sollte ein anschlagdynamisches 88er-Piano-Manual in rund 150 Mikrosekunden gescannt sein, das heißt, pro Taste bleiben dem Controller noch nicht einmal

Steckverbinder-Anschlussbelegungen

PL01 Netzteil	
1	AC 9 V oder DC 8 bis 12 V
2	AC 9 V oder DC 0 V Transformator 9V/4VA erforderlich
PL02 Pedal Switches	
1	P1.3 Aux/Sostenuto-Pedal
2	P1.4 Portamento-Pedal
3	P1.5 Sustain-Pedal
4	Gnd (Masse) Pedale schalten nach Masse
PL03 Atmel ISP-Interface	
1	MOSI
2	+5 V
3	nicht belegt
4	Gnd
5	Reset
6	Gnd
7	SCK
8	Gnd
9	MISO
10	Gnd
Belegung STK200- und PonyProg-kompatibel	
PL04 Manual Scan	
1	PB0 SRCLK (Schieberegister-Takt)
2	PB1 SRData (Schieberegister-Datenbit)
3	PB2 Ruhekontakt oder 1. Schließer
4	PB3 Arbeitskontakt oder 2. Schließer
5	PB4 MxRCLK Matrix-Reihen-Takt
6	PB5 Port B5 (nicht benutzt)
7	nicht belegt
8	nicht belegt
9	+5 V
10	Gnd
PL05 Analog I/O Interface	
1	PC0 Bend Wheel
2	PC1 SRData (Schieberegister-Datenbit)
3	PC2 Expression/Modulation Wheel
4	PC3 Analog Multiplexer/FootController
5	PC4 SDA I2C-Bus
6	PC5 SCK I2C-Bus
7	ADC6
8	ADC7
9	+5 V
10	Gnd
PL06 MIDI-Eingang	
1	MIDI+ (Pin 4 DIN-Buchse)
2	MIDI- (Pin 5 DIN-Buchse) für spätere Erweiterungen vorgesehen
PL07 MIDI-Ausgang	
1	MIDI+ (Pin 4 DIN-Buchse)
2	Gnd (Pin 2 DIN-Buchse)
2	MIDI- (Pin 5 DIN-Buchse)



Mit Ausnahme der Bedienelemente und des LCD werden alle Bauteile auf der SMD-Lötseite montiert. U05 kann direkt auf die Kupferfläche der Platine geschraubt werden, er benötigt dann keinen Kühlkörper. Bei den runden D6-Tastern gehört die abgeflachte Seite nach rechts.

Bedienungsanleitung

Bei der Firmware-Programmierung des MIDlvice wurde auf eine praxisingerechte Bedienung besonderer Wert gelegt – nicht umsonst belegen die User-Interface-Routinen rund 75 Prozent des mit 8 KByte nicht gerade üppigen ATmega8-Programmspeichers. Auch wenn wir im Artikel die „digitale“ Bedienung geißelten: Ganz ohne Knöpfchendrücken kommt man bei der Grundeinstellung nicht aus. Die umfangreichen Parameter sind auf insgesamt zehn LCD-„Bildschirmseiten“ verteilt, durch die man mit den Rechts/Links-Buttons blättert. Der angewählte Parameter wird durch einen „<-“ Cursor gekennzeichnet, er lässt sich sowohl schrittweise mit den Auf/Ab-Tasten als auch mit der „Select by Key“-Taste ändern. Drückt man selbige, leuchtet die LED im (oder am) Button, was zum Drücken einer dem gewünschten Zahlenwert zugeordneten Klaviatur-Taste auffordert. Parameter-Werte von 1 bis 61 (bei einem 5-Oktaven-Manual) lassen sich so unmittelbar aufrufen; für die meistbenutzten Parameter ist dieser Bereich völlig ausreichend. Ein Abbruch des Vorgangs ist durch nochmaliges Drücken des Tasters möglich. Hält man eine der Auf/Ab-Tasten dauerhaft gedrückt, verändert sich der Wert in

wachsender Geschwindigkeit, wie man es von einigen Digitaluhren kennt; auch das Blättern erfolgt mit „Auto-Repeat“.

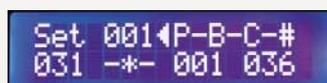
Jeder Wert kann natürlich nur in den jeweils möglichen oder sinnvollen Grenzen verändert werden, maximal im Bereich 0 bis 127 (MIDI-Datenbereich). Der Wert null wird als „*-“ dargestellt, was im MIDlvice-Kontext „nicht benutzt“ oder „nicht gesendet“ bedeutet. Die MIDI-Norm ist bei den Parametern leider etwas inkonsistent: Programmnummern und Kanäle etwa werden ab 1 gezählt, aber als Byte aufsteigend von 0 gesendet (angezeigter Wert –1). Demgegenüber beginnen Tastennummern sowohl bei der Übertragung als auch bei der Zählweise bei 0 (ganz, ganz tiefes „C“).

Die Parameter sind thematisch in Vierer-Gruppen zusammengefasst, wobei die ersten 32 Werte ein „Set“ (oft auch als „Patch“ bezeichnet) bilden, von denen es wiederum 16 gibt. Jedes Set parametrisiert für das verwendete Instrument die den analogen/digitalen Spielhilfen zugeordneten MIDI-Controller-Nummern. Den 32 Set-Parametern folgen zwei Seiten mit acht „Defaults“-Werten, die für alle Sets gemeinsam

gelten und praktisch die Grundeinstellung des Geräts beschreiben. Alle Werte speichert MIDlvice im ATmega8-EEPROM-Bereich, sie bleiben auch nach Abschalten der Versorgungsspannung erhalten.

Hauptseite

Nach dem Einschalten und der Begrüßungsmeldung präsentiert MIDlvice die Hauptseite mit Set 1. Der Cursor zeigt auf die Set-Nummer, und man kann eines der 16 Sets direkt anwählen.



In der unteren Status-Zeile zeigt das LCD die in diesem Set gespeicherte MIDI-Programmnummer und -Bank, den verwendeten MIDI-Channel und die MIDI-Basisnote des Manuals (36 = C). Bei der Auswahl eines Sets (und auch nach dem Einschalten für Set 1) werden MIDI-Programmnummer und -Bank an den PC beziehungsweise das virtuelle Instrument gesendet und Channel, Basisnote und MIDI-Controller ihren Werten entsprechend zugewiesen. Drückt man auf der Hauptseite den Links-Button, sendet MIDlvice die ana-

logischen/digitalen Eingangswerte sämtlicher zugewiesenen MIDI-Controller-Werte und zusätzlich einen All-Notes-Off-Befehl (Panik-Taste bei Tastenhängern). Das Instrument übernimmt dann alle aktuellen Regler- und Schalterstellungen, ohne dass man jedes einzelne Bedienelement „anfassen“ müsste.

Set-Parameter

Auf der ersten Parameter-Seite stellt man die Werte MIDI-Programmnummer und -Bank (Prg/Bnk, 1 bis 127), den verwendeten MIDI-Channel (Chn, 1 bis 16) und die MIDI-Basisnote des Manuals (Bn#, 12 bis 96) für das jeweilige Set ein. Der Wert „*-“ bedeutet bei Programm und



Bank, dass diese Informationen beim Wechseln des Sets nicht gesendet werden. Veränderte Programm- und Bank-Nummern werden unmittelbar übertragen, praktischerweise lässt man den Cursor während des Spiels einfach auf der Programmnummer.

Seite 2 zeigt die MIDI-Controller-Zuweisungen für das zweite

2 µs. In dieser Zeit muss er den Zustand der Kontaktfeder ermitteln (ruhend/unterwegs/angekommen), gegebenenfalls den zur Taste gehörenden Timing-Zähler inkrementieren und den Taktimpuls für die Scan-Platine generieren. Keine Frage, dass sich ein solches Problem nur in Maschinensprache (Assembler) lösen lässt. Auch wenn der kleine ATmega8 mit 16 MHz getaktet wird, wäre eine Hochsprache hier ungeeignet.

Die kommt allerdings für die „Hausaufgaben“, sprich die Menüführung und die MIDI-Aufbereitung zum Einsatz. Der Autor verwendete hier – mehr aus persönlichem Gusto – den Pascal-Cross-Compiler der Firma E-Lab [3]; die Entwicklung erfolgte auf dem kostengünstigen Mega8-Modulsatz von embedit [4]. Letztere Firma bietet für eine geringe Schutzgebühr auch eine auf 4 KByte Code beschränkte

Light-Version des E-Lab-Compilers an, der für eine reine MIDI-Tastaturabfrage mit Programmnummern-Auswahl völlig ausreicht. Ein leicht auf eigene Bedürfnisse anpassbares, funktionsfähiges Grundgerüst des Abfrage-Codes finden Sie auf unserer Projekt-Webseite (Soft-Link), fertig programmierte Mikrocontroller nebst Platinen gibt es wie immer bei eMedia. Die enthalten natürlich den vollen Funktionsumfang mit bequemer Menüführung über die kreuzförmig angeordneten Bedientaster.

Selbstgelötetes

Auch wenn nur handelsübliche Bauteile verwendet werden, sind bei der Bestückung der MIDlvice-Platine einige Besonderheiten zu beachten. Das MIDlvice ist gleichzeitig Bedieneinheit, Display und „Mainboard“,

wobei die Bedienelemente und das LCD auf der einen Seite, die restlichen Bauteile auf der anderen Seite montiert werden. Für den Mikrocontroller wählten wir die TQFP-Version, weil die Platzverhältnisse unter dem Display dies vorgaben und dieser Typ zwei analoge Eingänge mehr besitzt als die DIL-Ausführung. Mit 0,8 mm Pitch (Beinchen-Abstand) lässt er sich mit ruhiger Hand selbst verlöten; etwas Löt-sauglitze für eventuelle „Ausrutscher“ sollte aber trotzdem bereitliegen. Das Löten klappt mit etwas SMD-Flussmittelpaste, einem sehr spitzen LötKolben und 0,5-mm-Lötendraht am besten, wenn man den Baustein genau ausrichtet und zunächst an zwei diametralen Pins anheftet. Achten Sie unbedingt auf die richtige Polung, falsch herum eingelötete SMD-Bauteile lassen sich kaum zerstörungsfrei ausbauen. Etwas unkritischer ist

der I2C-Baustein zur Display-Ansteuerung zu verlöten, wählen Sie hier die größere SMD-Ausführung des PCA9555 mit dem Suffix D (1,27er Pitch). Eine DIL-Version des PCA9555 wird leider nicht hergestellt.

Das Display darf natürlich erst dann eingesetzt werden, wenn alle anderen Bauteile platziert und eingelötet sind, anderenfalls kommt man an einige Lötstellen nicht mehr heran. Als Display kann jedes zweizeilige LCD mit 16 Zeichen/Zeile verwendet werden, sofern es die äußeren Abmessungen 84 mm × 44 mm besitzt (Industrie-Standard); selbst die modernen OLED-Ausführungen sind geeignet. Unseren Prototypen haben wir mit dem elegant blau leuchtenden E162-NLW von Electronic Assembly bestückt. Es ruht auf 5-mm-Abstandsrollchen, seine LED-Hintergrundbeleuchtung wird über zwei zusätzliche Pins (15

Aux-Pedal (Digitalwert, neben dem fest vergebenen Sustain-Pedal, z. B. Sostenuto), den Status der vier externen Effektschleifen-Relais-Ausgänge (EfX, 0000 bis 1111 übersetzt in ihren De-

```
Aux EfX ExP Aft
-* 006 011 001
```

zimalwert 0 bis 15), den MIDI-Controller des Expression-Pedals/Modulationsrads (ExP) und den alternativen Aftertouch-Controller (Aft). Steht bei letzterem *- (null), verwendet MIDIvice den normalen MIDI-Aftertouch-Befehl, bei Werten ab 1 weist es dem Aftertouch-Sensor ebendiese MIDI-Controllernummer zu. So lassen sich mit der Aftertouch-Spielweise auch Instrumente beeinflussen, die gar keinen Aftertouch-Befehl akzeptieren.

Auf den Seiten 3 bis 6 legt man fest, welche MIDI-Controller den 15 Analogeingängen der I/O-

```
An1 An2 An3 An4
012 013 014 015
```

```
A13 A14 --- FcD
071 070 -* 007
```

und 16) über einen Vorwiderstand aus der Netzteil-Eingangsspannung (9 bis 12V) versorgt. Bei nicht beleuchteten Displays bleiben die Pads für Pin 15 und 16 leer. Etwas Augenmerk verlangt der Vorwiderstand R05, der entsprechend der Displaybeleuchtungs-Stromaufnahme zu bemessen ist. Unser oben erwähntes blaues Display benötigt 30 mA bei einer (im Display-Datenblatt angegebenen) LED-Durchlassspannung von 3,6 V, ergibt bei 12 V Netzteil-Eingang 270 Ohm. R05 sollte eine Leistung von 1 W aufweisen.

Die in fast jedem Elektronik-Bastelshop zu findenden runden D6-Printtaster sind beim Einsatz der etwas dicker aufragenden beleuchteten Displays leider nicht hoch genug, um noch durch die Frontplatte zu ragen. Alternativ lassen sich an dieser Stelle die größeren Multimec-Modultaster einsetzen, die mit

Platine zugewiesen werden sollen (An1 bis A14, FcD). Eingang 16 ist für das Lautstärke-Pedal reserviert, deshalb steht hier FcD (Foot Controller Destination). Eingang 15 liegt fest auf der Referenzspannung und wird nicht benutzt. Sind keine analogen Controller vorhanden, trägt man hier überall „*-“ (null) ein.

Seite 7 und 8 legen die Zuweisung der maximal acht Schalter (Sw1 bis Sw8) der I/O-Platine auf

```
Sw1 Sw2 Sw3 Sw4
030 031 066 067
```

ihre MIDI-Controller fest. Ein geschlossener Schalter sendet den Datenwert 127, ein offener eine 0 – natürlich (wie bei den analogen Controllern) nur bei einer Änderung.

Default-Parameter

Auf den zwei letzten Seiten findet man die MIDIvice-Default-Einstellungen, die für alle Sets gleichermaßen gelten. Dyn wählt eine von zwei exponentiellen Tastendynamik-Kurven aus (1 und 2, unterschiedliche Steigungen), stellt eine zeitlineare (3, eher unvorteilhaft, nur

für Sonderfälle) oder gar keine Tastendynamik (-*) ein. In letzterem Fall wird der unter MnD (MIDI Note On Dynamik) eingestellte Wert für die Anschlagstärke gesendet. Den angewählten Wert MnD kann man durch mehr oder weniger starkes Anschlagen einer einzelnen Taste direkt einstellen; auf diese Weise lässt sich auch feststellen, ob das mit Tim gewählte Ab-

```
Dyn Tim MnD AcS
001 060 064 -*
```

frage-Timing zur Mechanik der Tastatur passt. Nimmt man den Timing-Wert mal vier, erhält man die Abfrage-Zykluszeit und damit die zeitliche Dynamik-Auflösung in Mikrosekunden. Letztendlich lässt sich nur durch Ausprobieren ausgehend vom Standardwert 240 µs (60) ermitteln, welche Einstellung ein optimales Anschlagsgefühl ergibt. AcS schließlich schaltet das MIDI Active Sensing ein (1) oder aus (-*).

Mit den drei letzten Default-Parametern legt man die Eigenschaft der verwendeten Tastatur fest. Ks# repräsentiert die Anzahl der abzufragenden Tasten inklusive

des Offsets (nächster Parameter). KbO (Keyboard Offset) stellt einen generellen Klaviatur-Versatz ein, wenn die Tastatur nicht mit einem „C“ beginnt (z. B. Klavier, Spinett-Orgeln mit 3,5 Okta-

```
Ks# KbO MxC ---
049 -* 008 -*
```

ven) oder eine Keyboard-Matrix mit der „falschen“ Note anfängt (zum Beispiel bei Yamaha-Tastaturen). Der letzte Parameter MxC ist nur für Matrix-Tastaturen relevant (Firmware-Version „Matrix“), er gibt an, wie viele Spalten eine Matrix-Reihe ergeben (meist 6 oder 8).

Vorgaben für alle Sets und die Default-Werte können übrigens schon bei der Flash-Programmierung des Mikrocontrollers definiert werden, indem man dem Programmier ein EEPROM-HEX-File mit den gewünschten Werten anliefern. Das recht triviale Datenformat findet sich als structconst-Definition im Controller-Sourcecode. Alternativ kann man sich auf Set 1 eine eigene Vorgabe zusammenstellen; hält man beim Einschalten den „Select by Key“-Taster gedrückt, kopiert MIDIvice die Werte aus Set 1 in alle anderen Sets.

verschieden hohen Tastenkapfen auf dem Markt sind; zusätzliche Bohrlöcher auf der Platine sind dafür vorhanden. Wir verwendeten für den „Select by Key“-Taster (siehe Kasten „Bedienungsanleitung“) eine transparente Kappe und fädelten eine blaue LED in den MultiMec-Tastenkörper ein (Polung beachten, quadratisches Lötpad = Kathode). Im Schaltbild sind zwei LEDs für die normale und die im Taster integrierte Bestückungsposition eingezeichnet, in Wirklichkeit handelt es sich aber nur um eine einzige LED.

Selbstverständlich kann man Display und Taster auch getrennt von der MIDIvice-Platine einbauen und über kurze Litzen (Flachbandkabel des Displays max. 15 cm) anschließen. Für die Anschlüsse der Peripherie-Platinen und eventuell des Flash-Programmers kommen 10-polige doppelreihige Pfostenstecker, für

die MIDI-, Netzteil- und Pedal-Anschlüsse einreihige Platinenverbinder zum Einsatz. Mit dem Trimpoti stellt man einmalig den Display-Kontrast ein, für eventuelle Korrekturen sollte der Trimmer halbwegs zugänglich bleiben. Auf der Platine ist ein Optokoppler für den MIDI-Eingang vorgesehen, der im vorliegenden Anwendungsfall aber (noch) nicht benötigt wird. Wenn Sie eigene Firmware-Entwicklungen wagen wollen, kann ein MIDI-Eingang aber ganz nützlich sein, etwa für eine MIDI-Merge-Funktion.

Nach sorgfältiger Sichtprüfung mit einer Lupe auf verirrte Lötperlen und Kurzschlüsse kann die Platine testweise in Betrieb genommen werden, in dem man eine Spannung von 9 bis 12 V an die Netzteil-Pins anlegt. Der (programmierte) Mikrocontroller sollte eine Begrüßungsmeldung zeigen und dann in den „Set“-

Auswahlmodus gehen, gegebenenfalls kann man am Trimpoti R07 noch den Display-Kontrast nachregeln. Auch ohne angeschlossene Scan- und I/O-Platine können Sie nun die Bedienung üben, denn diese Peripherie stellen wir erst in den nächsten Heften vor. (cm)

Literatur

- [1] Carsten Meyer, Zurück zur Natur, Mit dem Apple II auf Klangfang, c't 12/84, S. 42
- [2] MIDI-Kurzkompendium, www.zem-college.de/midi/index.htm
- [3] E-Lab AVR Embedded Pascal, www.e-lab.de
- [4] Atmega8 Minimodul Application Board und ISP-Adapter, shop.embedit.de
- [5] PonyProg, Lanconelli Open Systems, www.lancos.com/prog.html