

MIDI_COM

LOW-COST Netzwerk-Alternative

Autor: Harald Blees
entwickelt mit: PURE-C

Bitte beachten Sie:

1. Dieses Handbuch ist - ebenso wie das Programm selbst - rechtlich geschützt. Es darf weder insgesamt noch teilweise kopiert werden. Alle Rechte liegen beim Autor.
2. Der Autor übernimmt keine vertraglichen Verpflichtungen gegenüber dem Käufer oder Anwender der Software. Er weist vorsorglich auf folgendes hin:
 - a) Dieses Handbuch enthält eine anwenderorientierte Beschreibung der Software, jedoch keine Garantien für bestimmte Eigenschaften oder Einsatzerfolge
 - b) Es ist nach dem Stand der Technik nicht möglich, Programme so zu erstellen, daß sie für alle Anwendungsbedingungen fehlerfrei sind.

ERSTELLEN SIE BITTE NUR KOPIEN FÜR IHRE EIGENEN RECHNER !!!

Allgemeines	3
Installation	4
MIDI_COM installieren	4
Multi-TOS	5
MagiC	6
Auflösungswechsel	6
Das MagiC_XFS	6
MagiC-Patch	7
MIDI-Schnittstelle	8
MIDI-Treiber	9
CCN-LIGHT (ROM-Port)	10
ROMPORT-Treiber	10
Multi-Schnittstellen Treiber	11
Sockets	12
MC_CONTR.PRG	13
Grundparameter	14
Netzparameter	15
Grunddialog Druckerverhalten	16
Druckerauswahl	17
Das Netzlaufwerk	18
Zugriffsschutz	18
SICHERHEIT	19
Die Druckumlenkung	20
Utilities	
MICO.CPX	22
MC_STOP/MC_START	23
MC_WATCH	23
Einschränkungen	24
Hilfe	25
Laser-Drucker	25
Drucken	25
Programmeinbindung	26
Grundlagen	26
Vereinbarungen	27
Abmelden	28

Ringkarte anfordern	29
Statusbits	29
Daten senden	29
Adressierung	30
Daten Empfangen	31
Programmbeispiel	32
Das Treiber-Konzept	33
Treiberfunktionen	33
Strukturen	35

Allgemeines

MIDI_COM ist ein einfaches Netzwerk, das bis zu 7 ATARI-Computer über die MIDI-Schnittstelle verbindet. Einfach daher, da es einige netzwerktypische Funktionen wie File- und Recordlocking nicht anbietet und außerdem nur GEMDOS-Funktionen unterstützt. Allerdings bietet MIDI_COM einige Besonderheiten, die ich bisher bei keinem anderen mir bekannten MIDI-Netzwerk gefunden habe.

Zum einen ist es mit MIDI_COM möglich, auf mehr als ein Fremdlaufwerk zur gleichen Zeit zuzugreifen. (Maximal 20 geöffnete Files auf einer beliebigen Kombination von Fremdlaufwerken.) Außerdem kann von allen Rechnern gleichzeitig auf ein einziges Laufwerk geschrieben werden, ohne das man in Gefahr läuft den Datenträger zu zerstören. Bei gleichzeitigem Schreiben in das gleiche File (an die gleiche Stelle) gewinnt immer der letzte Schreiber, ansonsten entstehen auch dabei keine Probleme.

Eingebaut ist zusätzlich eine Umlenkung der Druckausgaben, so daß eine verteilte Nutzung von Druckern durch alle im Netz eingebundenen Rechner ermöglicht wird.

Installation

Um das Netz nutzen zu können, müssen alle Rechner miteinander über eine Schnittstelle verbunden werden. Auf der Diskette finden Sie standardmäßig Treiber, die MIDI_COM für eine Verbindung über die MIDI-Schnittstelle benötigt, sowie für eine Verbindung über die ROM-PORT-Hardware CCN-LIGHT. Weiterhin wurde ein Treiber für die serielle Schnittstelle von Harun Scheutzow zur Verfügung gestellt, dem ich hiermit meinen herzlichen Dank aussprechen möchte.

Es ist zu beachten, das für TOS-Versionen bis einschließlich TOS 1.4 das Programm INSJAR.PRG vor den Schnittstellentreibern im Autoordner installiert sein sollte, da die Schnittstellentreiber einen installierten COOKIE-JAR voraussetzen.

MIDI_COM installieren

Grundsätzlich muß MIDI_COM auf allen Rechnern im Ring gebootet werden. Dazu kopieren Sie am besten MIDI_COM.ACC und MIDI_COM.INX auf Ihre Bootpartition sowie den gewünschten Schnittstellentreiber in den Autoordner.

Die Programme im Ordner MC_UTILS sowie das Demoprogramm MC_TALK.ACC auf der Diskette sind nicht für den normalen Betrieb des Netzwerkes notwendig.

Multi-TOS

Um MIDI_COM unter Multi-TOS zu betreiben muß zusätzlich zu den oben beschriebenen Files das spezielle eXtended-File-System (MCXFS.XFS) sowie MC_MINT.PRG aus dem Ordner Multi-TOS Ihrer Originaldiskette in Ihren Multi-TOS-Ordner kopiert werden. Unter Multi-TOS ist MIDI_COM zur Zeit auf das Netzlaufwerk N:

festgelegt. MIDI_COM ignoriert in diesem Fall das in MIDI_COM.INX gewählte Netzlaufwerk.

Ergänzen Sie nun noch MINT.CNF um den Eintrag:

```
exec /Multi-TOS/MC_MINT.PRG
```

und ändern Sie MIDI_COM.ACC auf Ihrer Bootpartition in MIDI_COM.PRG um. Ein eventuell eingestellter ACC-Pfad wird nicht berücksichtigt.

Leider kann sich MIDI_COM zur Zeit noch nicht mit den Speicherschutzmechanismen von MiNT anfreunden. Aus diesem Grund muß MINT.PRG in MINTNP.PRG umbenannt werden, falls es auf einem FALCON oder TT zusammen mit MIDI_COM eingesetzt werden soll.

Sollen Auflösungswechsel vorgenommen werden, ist es unter Multi-TOS notwendig, MIDI_COM vor einem Auflösungswechsel zu beenden, indem man z.B.: mit PS_CONTR ein KILL-Signal an MIDI_COM sendet.

Um eine möglichst saubere Umlenkung der Druckerausgaben zu erreichen, sollte in MINT.CNF die folgende Zeile ergänzt werden:

PRN=U:/PIPE/MICO

MagiC

Die Installation unter MagiC ist genauso wie unter TOS vorzunehmen.

Zusätzlich kann unter MagiC ab Version 3.00 ein eXtended-File-System(XFS) installiert werden.

Das Programm MagC_XFS.PRG im Ordner MagiC auf Ihrer Diskette muß vor MIDI_COM gestartet werden. Man kann es sowohl im Autoordner als auch im Applikations-Ordner unterbringen. Der Applikations-Ordner wird in MagiC.INF unter

#_APP festgelegt.

Damit das Filesystem auch unter MagiC 3.00 korrekt läuft muß zusätzlich noch eine kleine Änderung in MagiC gepatcht werden. Neuere MagiC-Versionen laufen inzwischen weitgehend problemlos.

Auflösungswechsel

Nicht in den AUTO-Ordner, sondern in den Applikations-Ordner geht das Programm MC_RESC.PRG. Dieses Programm dient dazu, MIDI_COM einen SHUTDOWN unter MagiC mitzuteilen, sodaß sich MIDI_COM korrekt beenden kann. Speziell beim Ändern der Auflösung ist dieses Programm unbedingt notwendig, da es sonst zu Abstürzen kommt.

Unter TOS erkennt MIDI_COM einen Auflösungswechsel selbsttöndig.
Das MagiC_XFS

Die Verwendung des eXtended File Systems unter MagiC ist für den normalen Gebrauch von MIDI_COM unter MagiC nicht unbedingt nötig.

Zur Zeit unterscheidet sich der Betrieb mit XFS nur durch:

1. U:\N\.. ist verfügbar
2. Programmstarts vom Netzlaufwerk laufen sauber.
Mit XFS startet MagiC ein Programm selbst, andernfalls wird der Programmstart durch MIDI_COM simuliert, was nicht ganz sauber funktioniert, dem normalen Anwender aber nicht auffällt.
3. Aufrufe des Netzlaufwerks laufen deutlich schneller ab.

Das eigentliche Ziel des XFS ist jedoch, den Filetransfer über MIDI_COM in den Hintergrund zu legen, sodaß mit anderen Programmen weitergearbeitet werden kann. Dies wird noch eingebaut, sobald das XFS vollständig ist.

MagiC-Patch

Zusammen mit dem Original-MagiC Version 3.xx kommt es bei Zugriffen auf das Netzlaufwerk über öU:\N\...ö zu abstürzen.

In MagiC ab Version 4.01 ist dieser Fehler bereits behoben!!

Filegr"Be MAGIC.RAM Version 3.xx: 168184

Das Patchprogramm nur anwenden, wenn Ihr MagiC GENAU diese Länge hat!!

Ohne den Patch sollten Sie daher das MagX_XFS.PRG nicht installieren. MIDI_COM ist dann nur direkt über N:\ ansprechbar (oder eben das Netzlaufwerk, das Sie eingestellt haben.) Dieser Fehler wird mit dem beileigenden Patch korrigiert. Hat Ihr MAGIC.RAM eine andere Länge als oben angegeben, wenden Sie sich bitte an den MagiC-Autor. Er ist über den Fehler informiert.

Kopieren Sie Ihr Original MAGIC.RAM zusammen mit MAGICPAT.PRG in einen Ordner.

Starten Sie jetzt MAGICPAT.PRG. Nun wird MAGIC.RAM in diesem Ordner gepackt. Anschliessend k"nnen Sie das MAGIC.RAM aus diesem Ordner wieder auf Ihre Boot-Partition kopieren.

Das Patchprogramm liegt auch als Source auf der Diskette vor.

Wenn Fehler auftreten sollten, k"nnen Sie ja jederzeit auf das Original MAGIC.RAM auf Ihrer Originaldiskette zurückgreifen.

MIDI-Schnittstelle

Um die Verbindung über die MIDI-Schnittstelle zu installieren, werden nur normale 3-adrige Kabel mit DIN-Steckern benötigt. Da die handelsüblichen Leitungen aber oft nicht die benötigte Länge aufweisen, wird man in den meisten Fällen dazu gezwungen sein, sich selbst die passenden Verbindungen anzufertigen. Die Pinbelegung der Stecker kann dem Handbuch zum ATARI Kapitel "Schnittstellen" entnommen werden. Im Fachhandel kann man sich meistens die Stecker gleich an die Kabel löten lassen.

Eine maximale Länge der Verbindungskabel ist mir bisher nicht bekannt. Von mir wurden Kabellängen von etwa 50 Metern getestet. Die Obergrenze dürfte aber höher liegen, da die MIDI-Schnittstelle nach dem Stromschleifenprinzip arbeitet und nicht mit Spannungspegeln, wie die anderen Schnittstellen.

Die korrekte Verbindung wird hergestellt, indem man den MIDI-OUT Ausgang des ersten Rechners mit dem MIDI-IN Eingang des nächsten Rechners verbindet. Dann dessen MIDI-OUT Schnittstelle mit der MIDI- IN Buchse des dritten und so fort, bis dann der letzte MIDI-OUT Anschluß wieder mit dem Eingang des letzten Rechners verbunden wird. Es hat sich als sinnvoll erwiesen, jedes Kabel auf einer Seite mit IN und auf der anderen Seite mit OUT zu beschriften, um Verwechslungen zu vermeiden.

Die Übertragungsgeschwindigkeit des Netzwerkes ist sehr stark abhängig von dem verwendeten Medium. Der Zugriff auf eine externe Harddisk ergab bei zwei Rechnern einen Durchschnitt von ca. 2000 Bytes pro Sekunde. Bei einer hochformatierten Diskette fiel der Durchschnitt auf etwa 1000 Bytes pro Sekunde ab. Von einer normalformatierten Diskette kann man etwa 1200 Bytes/Sek. erwarten. Diese Werte wurden handgestoppt und gelten sowohl bei Lese- als auch Schreiboperationen über den Ring (ohne CACHE-Programme).

MIDI-Treiber

Bei Verwendung der MIDI-Schnittstelle sollte man das Programm MC_RESET.PRG an erster Stelle in den Auto-Ordner kopieren. Dieses Programm verhindert bei späteren Boot-Vorgängen, daß eingehende MIDI-Daten die Interruptlogik des Atari durcheinander bringt.

Unter verschiedenen Umgebungen scheinen einige Tricks, die der Treiber verwendet um die MIDI-Schnittstelle zu stabilisieren, zu Problemen zu führen. Daher hier eine Sammlung mit verschiedenen Methoden. Probieren geht über studieren. Auf 16 .. 32 Mhz-Rechnern kann man MC_DRV_E.PRG installieren. Auf allen anderen sollte man es zuerst mit MC_DRV_C.PRG oder MC_DRV_D.PRG probieren.

Die Aufstellung zeigt, welche Interruptvektoren der jeweilige Treiber zusätzlich zum MIDI-Vektor verändert, um die MIDI-Schnittstelle auf Empfangene Daten zu überprüfen.

Die Änderung des Interrupt-Levels ist ein Trick, der es dem Prozessor erlaubt, auch während VDI-Aufrufen MIDI-Interrupts bearbeiten zu können. Leider funktioniert diese (effektive) Technik nur auf 68000-Prozessoren.

TREIBER	IKBD	MAUS	Interrupt-Level
MC_DRV_A.PRG	X	X	X
MC_DRV_B.PRG		X	X
MC_DRV_C.PRG		X	
MC_DRV_D.PRG			X
MC_DRV_E.PRG			
MC_DRV_F.PRG	X	X	

Empfehlungen:

TT	=>	MC_DRV_E
FALCON	=>	MC_DRV_E oder MC_DRV_C
MegaSTE	=>	MC_DRV_C oder MC_DRV_D
andere	=>	MC_DRV_B, MC_DRV_C oder MC_DRV_D

Die Treiber A, B und D dürfen nur bei 68000-CPU verwendet werden, ansonsten kommt es in unregelmäßigen Abständen zu Abstürzen!

CCN-LIGHT (ROM-Port)

Seit einiger Zeit gibt es eine Hardwareergänzung, mit deren Hilfe zwei Atari-Computer über den ROM-PORT verbunden werden können. Die Bezugsadresse erfahren Sie bei Ihrem Händler.

Die Übertragungsgeschwindigkeit mit den beiliegenden Treibern für MIDI_COM ist direkt abhängig von der Geschwindigkeit der verbundenen Rechner. Zwei normale ST's (8 MHz) mit einer MEGAFILE30 Harddisk erreichen etwa 32 KBytes/Sekunde.

Die Installation dieser Verbindung ist denkbar einfach. Als erstes stecken Sie die ROM-PORT Verbindung in Ihre Rechner.

Diese sollten dazu vorher etwa 5 Minuten lang ausgeschaltet sein.

Anschließend starten Sie beide Rechner und testen mit dem Programm CCNINIT.PRG den korrekten Sitz der Stecker.

ROMPORT-Treiber

Das Programm CCN-INIT.PRG sollte möglichst am Anfang des Autoordners stehen. Dieses Programm initialisiert den ROM-Port für den Einsatz des Treibers.

Auf der Diskette finden Sie im Ordner TREIBER\AUTO.CCN die notwendigen Treiber, die MIDI_COM für die Hardwareerweiterung CCN-LIGHT von Christian Krüger benötigt.

Die Treiber für das Romport-Modul legen nicht automatisch die Rechnernummern fest. Daher muß ein Rechner MC0NCC04.PRG benutzen und der andere MC1NCC04.PRG.

Zusätzlich wurden Treiberpaare ergänzt, welche die in MIDI_COM neu eingebaute Blockgrößen-Steuerung verwenden. Dazu muß natürlich (durch den Treiber) zusätzlicher Speicher angefordert werden. Dafür erhöht sich die Übertragungsgeschwindigkeit aber erheblich. Die Treiber befinden sich im Verzeichnis CCN_FAST

Sie haben die Wahl zwischen 16KB-Blöcken oder 30KB-Blöcken.

Es ist nicht möglich, die Treiber für Standardblöcke mit den erweiterten Blockgrößen gemischt zu verwenden.

Multi-Schnittstellen Treiber

Da der ROM-Port bisher nur zwei Rechner verbinden kann, wurde auf mehrfachen Wunsch nun auch ein Treiber für den gemischten Betrieb der verschiedenen Schnittstellen entwickelt.

Da es sich bei diesem Treiber um eine Neuentwicklung handelt, sind kleine Anfangsprobleme sicherlich nicht auszuschließen. Fehlermeldungen bitte ruhig an den Autor. Die Adresse finden Sie im Hypertext auf Ihrer Originaldiskette.

Zur Installation benötigen Sie die Programme MULTI_0.PRG bis MULTI_2.PRG. Diese werden ganz normal im Auto-Ordner installiert. Zusätzlich müssen Sie nun noch ein Verzeichniss SOCKETS im Auto-Ordner anlegen. In diesen kopieren sie dann die gewünschten Schnittstellenserver. Für weitere Rechner fragen Sie bitte den Autor nach passenden MULTI_x-Treibern. Diese werden Ihnen dann Kostenlos zugeschickt.

Die Reihenfolge der Socket-Programme im SOCKETS-Ordner kann durchaus eine Rolle bei der Endgeschwindigkeit spielen. Die Sockets werden durch MIDI_COM in der Reihenfolge bearbeitet, in der sie physisch im Ordner stehen.

Der Mischbetrieb MIDI/SERIELL dürfte höchstens auf einem TT halbwegs zufriedenstellend laufen. Der gleichzeitige Betrieb der beiden Schnittstellen auf langsameren Rechnern ist erfahrungsgemäß nicht sinnvoll

Diese neuen Treiber haben ein eigenes (neues) internes Protokoll. So wird nun auf den TOKEN auf der MIDI-Schnittstelle verzichtet. Außerdem werden keine CRC-Prüfsummen berechnet. Dies führt auf der MIDI-Seite zu einem Geschwindigkeitzuwachs von etwa 300 Bytes/Sek.

Die ROM-Port-Seite entspricht in der Geschwindigkeit etwa dem 4KB-Treiber. Durch den erhöhten Verwaltungsaufwand verliert der ROM-Port jedoch ein paar KB/Sek. Dafür können nun wieder von jedem Rechner aus alle Ressourcen genutzt werden.

Sockets

Serielle Schnittstelle

Um diese zu verwenden ist die Installation des HSMOD-Paketes notwendig. Es ist darauf zu achten, dass die Puffergrößen (Senden und Empfang) im SCC.PRГ (für SERIEL2) bzw. MFP.PRГ (MODEM1) jeweils auf mindesten 10000 Bytes eingestellt sind. Bei mir stehen beide auf 15000 Bytes.

Socket-Programm	Device	Baud	Kommunikation
SO_M119.PRГ	U:\DEV\MODEM1	19200	8/1N no Handshake
SO_M1115.PRГ	U:\DEV\MODEM1	115200	8/1N no Handshake
SO_S219.PRГ	U:\DEV\SERIELL2	19200	8/1N no Handshake
SO_S2115.PRГ	U:\DEV\SERIELL2	115200	8/1N no Handshake

Beachte: MODEM1 ist auf ATARI's normalerweise nur bis 19200 Baud konfigurierbar.
Unter MagiC-PC kann dieser Socket jedoch mit 115200 Baud verwendet werden.

MIDI-Schnittstelle

SO_MIDI.PRГ direkter Zugriff auf die MIDI-Schnittstelle ohne Optimierungen

ROM-Port

SO_CCN.PRГ Der ROM-Port arbeitet hier nur mit der Standard Blockgröße von 4KB, da die anderen Schnittstellen (speziell MIDI) mit größeren Blöcken Schwierigkeiten haben.

Beispiel:

Rechner A <--MIDI--> Rechner B <--ROM-Port--> Rechner C
Installation auf Rechner A

AUTO\MULTI_0.PRГ

AUTO\SOCKETS\SO_MIDI.PRГ

Installation auf Rechner B

AUTO\MULTI_1.PRГ

AUTO\SOCKETS\SO_MIDI.PRГ
AUTO\SOCKETS\SO_CCN.PRГ

Installation auf Rechner C

AUTO\MULTI_2.PRГ

AUTO\SOCKETS\SO_CCN.PRГ

MC_CONTR.PRG

Das Programm MC_CONTR dient zur Änderung und Einstellung der Parameter von MIDI_COM sowie deren Abspeicherung in der Datei MIDI_COM.INX. Ab Version 3.92 liest MIDI_COM die binäre Datei MIDI_COM.INX anstelle der ASCII-Datei MIDI_COM.INF. MC_CONTR lößt sich sowohl als Programm als auch als ACC einsetzen. Hierzu muß nur die Endung PRG in ACC umgeändert werden.

Nach Programmstart oder Auswahl des ACC-Eintrages erscheint zuerst die Fileselectorbox, zur Auswahl des zu ändernden MIDI_COM.INX-Files. Es werden aber nur die nicht-Online Parameter übernommen. Alle anderen Werte kommen direkt aus dem aktiven MIDI_COM. Anschließend erscheint die Hauptauswahl in der Mitte des Bildschirms:

Die einzelnen Einstelloptionen werden mit der Maus durch einen Click auf das entsprechende Icon ausgewählt. Jedes der geöffneten Fenster kann jederzeit geschlossen werden. Erst durch schließen des letzten offenen Fensters wird das Programm verlassen.

In jedem Einstell-Dialog finden Sie die Buttons:

Bei Anwahl wird das Hauptauswahlmenü in den Vordergrund gebracht. Wurde das Fenster geschlossen, wird es erneut geöffnet.

Die vorgenommenen Änderungen und Eingaben werden aus dem Dialog übernommen. Ausser bei den Boot-Parametern werden die getroffenen Einstellungen auch sofort von MIDI_COM übernommen.

Schließen aller offenen Fenster, keine Sicherung in MIDI_COM.INX

Sichern der Einstellungen in MIDI_COM.INX

Alle Werte müssen vorher mit Aktivieren übernommen worden sein. Andernfalls werden nicht die neuen Werte gesichert. Eine UNDO-Funktion wird in zukünftigen Versionen folgen.

Sichern der Fenster-Positionen und Konfiguration von MC_CONTR.

Sind außer dem Auswahlfenster noch weitere geöffnet, wird beim Neustart das Auswahlfenster nicht wieder geöffnet.

Grundparameter

Dieses Icon steht für die grundlegenden Netzparameter, wie etwa die Festlegung des Netzlaufwerkes. Diese werden in der Bootphase durch MIDI_COM eingestellt und lassen sich nicht Online verändern. Die hier vorgenommenen Einstellungen werden also erst nach dem nächsten Booten aktiv.

Die Eingabe unter Node-Name bezeichnet den Namen, unter welchem der Rechner im Netzwerk bekannt ist.

Das Netzlaufwerk bestimmt das Laufwerk auf dem Desktop, welches durch MIDI_COM verwendet werden soll.

Ist Systemzeit setzen aktiviert, wird von diesem Rechner aus die Uhrzeit und das Datum beim Booten an die anderen Rechner verteilt. Diese Option sollte nur auf einem Rechner im Netz aktiv sein.

Die Größe des Spool-Puffers wird in BYTES angegeben. Die Eingabe 0 bedeutet, dass der Spooler nicht aktiv ist und MIDI_COM die Druckausgaben direkt über das GEMDOS vornimmt. Dies ist unter anderem bei allen Druckern sinnvoll, die nicht über den Parallelport angeschlossen sind.

Außerdem erlaubt diese Option die Verwendung anderer Spooler. Allerdings sind nicht alle Spooler geeignet. Ein wichtiges Kriterium ist, dass der externe Spooler nicht ebenfalls das Betriebssystem zur Druckausgabe verwendet.

Als weitere Option kann bei Einstellung des Spoolers entschieden werden, ob der notwendige Speicher am Anfang oder am Ende des Hauptspeichers reserviert werden soll. Eine Reservierung am Ende des Hauptspeichers (Pfeil abwärts) verhindert eine Zerstückelung des Hauptspeichers, verursacht aber leider unter bestimmten Bedingungen (Mag!X, FALCON-Auflösungswechsel) einige Probleme.

Netzparameter

Einstellung der Netzparameter. (Online)

Der Wert mcspeeder bestimmt die Pausen die MIDI_COM einlegt. Dadurch ist es m'glich, die Belastung ihres Rechners durch das Netzwerk den eigenen Bedürfnissen anzupassen. Unter Multi-TOS sollte dieser Wert niemals 0 sein.

Die Fehlerkontrolle wird zur Zeit nur durch den MIDI-Treiber unterstützt. Wird die Kontrolle aktiviert, werden am oberen Bildschirmrand die Aktivitäten im Netz angezeigt.

Ebenfalls zuschaltbar ist die TOS-Kompatibilität. Mit dieser Option kann MIDI_COM auch dann arbeiten, wenn nicht GEM Programme wie etwa GFA-Basic laufen. Diese Option funktioniert im allgemeinen aber nur, wenn diese Programme regelmäßig die Tastatur über das Betriebssystem auf Eingaben abfragen.

Grunddialog Druckerverhalten

Über dieses Icon erreichen Sie den Grunddialog für das Druckerverhalten. (Online)

MIDI_COM speichert Druckausgaben zuerst in einem internen Puffer, um bei Netzbetrieb eine möglichst effiziente Übertragung zu gewährleisten. Die Blockgröße bestimmt den Schwellwert, ab wieviel Bytes eine Ausgabe an den Drucker (über das Netz oder direkt) erfolgen soll.

Da eine Druckausgabe im allgemeinen nicht genau durch den Schwellwert teilbar ist, muß irgendwann auch der Rest ausgegeben werden. Zu diesem Zweck dienen die beiden timeout-Werte.

Diese bestimmen die ungefähre Zeit, nachdem MIDI_COM den internen Puffer ausgibt, ohne daß der Schwellwert erreicht wurde (timeoutxx). Wurde als letztes Zeichen ein Formfeed (Seitenvorschub) an den Drucker gegeben, so liegt die Wahrscheinlichkeit, daß keine weiteren Zeichen folgen werden natürlich höher. Daher kann für diesen Sonderfall die Zeitgrenze extra eingestellt werden (timeoutFF). Leider wird durch die meisten Programme noch eine Steuersequenz angehängt, sodaß dieser Parameter in den meisten Fällen bedeutungslos ist.

Da ein Drucker von mehreren Rechnern im Netz verwendet werden kann, erzeugt MIDI_COM nach erfolgtem Timeout einen Seitenvorschub, falls das letzte ausgegebene Zeichen kein Formfeed war. Dieses Verhalten kann über Option FFö aktiviert werden.

Die Druckumlenkung kann auch ganz abgeschaltet werden. Ohne Druckumlenkung werden Druckausgaben von MIDI_COM nicht mehr abgefangen. Allerdings kann dadurch zu Problemen kommen, wenn zeitgleich von anderen Rechnern im Netz auf diesen Drucker ausgegeben wird.

Aus diesem Grund ist es zusätzlich möglich, den Drucker zusätzlich für MIDI_COM zu sperren. In diesem Fall ist der angeschlossene Drucker für alle anderen Rechner im System unsichtbar. Außerdem kann man mit dieser Option auch spezielle Geräte am Parallelport verwenden, wenn diese direkt und nicht über das GEMDOS angesprochen werden. Ausdrücke kann man so über MIDI_COM an einen anderen Rechner umlenken und Plotterausgaben (falls die Software mitspielt) direkt auf den Parallelport ausgeben, ohne daß sich MIDI_COM einmisch.

Als letztes kann auch das Senden des Testzeichens durch MIDI_COM bei jedem Druckbeginn beeinflusst werden. Bisher h"rte man bei den meisten Druckern eine Piepton bei jedem Duckstart. MIDI_COM pr"uft damit die Ausgabebereitschaft des Druckers. Da dieses zus"tzliche Zeichen auch bei l"ngeren Pausen eingef"ugt werden kann (timeout), kam es bei Grafikdrucken zeitweise zu Problemen. Nun kann man sich hier aussuchen, ob ein Testzeichen benutzt werden soll, und wenn ja auch welches. Das Zeichen 070 steht f"ur BELL, erzeugt also den beliebten Piepton.

Druckerauswahl

Auswahl der erlaubten Zieldrucker

Bei einer Druckausgabe sucht MIDI_COM einen empfangsbereiten Drucker im Netz. Als erstes wird gepr"uft, ob am ausgebenden Rechner direkt ein Drucker angeschlossen und ausgabebereit ist. Ein Drucker ist nicht ausgabebereit, wenn er entweder gerade durch einen anderen Netzrechner verwendet wird oder aber nicht Online geschaltet ist.

Die Suche nach einem Drucker kann zus"tzlich eingeschr"nkt werden. MIDI_COM ber"cksichtigt bei der Suche nur solche Drucker, die "ber diesen Dialog als erlaubt gekennzeichnet sind. Die Einstellungen werden sofort aktiv und k"nnen auch permanent im .INX-File hinterlegt werden. Allerdings k"nnen nur die Rechner ausgew"hlt werden, die Online erreichbar sind. Diese Auswahl l"st sich daher nur sinnvoll einstellen, wenn das Netzwerk bereits aktiv ist.

Des weiteren mu" bemerkt werden, das es mit dieser Auswahl nicht m"glich ist, mehrere Drucker von einem Rechner aus gleichzeitig zu beschicken. MIDI_COM sucht erst nach einem neuen Drucker, wenn ein Druckauftrag beendet worden ist. Nur mit Hilfe von (gro"en) externen Spoolern k"nnte man einen entsprechenden Effekt erreichen. Der eingebaute Spooler jedoch verhindert eine Freigabe bis das letzte Zeichen auf dem Drucker ist.

Das Netzlaufwerk

Ist der Ring geschlossen und das Netzlaufwerk (normalerweise öN:ö siehe auch MIDI_COM.INX) auf dem Desktop installiert, so kann man über dieses Laufwerk auf alle anderen Laufwerke im Ring (ausser den eigenen) zugreifen. Dazu zeigt MIDI_COM für jeden angeschlossenen Rechner im Ring einen Ordner öUSERö an.

Anstelle der Bezeichnung öUSERö kann in MC_CONTR auch ein eigener Name angegeben werden (Node-Name), unter dem dann der jeweilige Rechner von den anderen Teilnehmern angesprochen werden kann.

Bei Namensgleichheit ersetzt MIDI_COM automatisch den letzten Buchstaben des Namens durch seine interne Adresse um eine eindeutige Identifizierung der Rechner über diesen Namen zu ermöglichen.

In jedem dieser Ordner befindet sich für jedes auf dem entsprechenden Rechner angemeldete Laufwerk wiederum ein Ordner. So kann man etwa mit dem Pfad öN:/USER/Ax/DESKTOP.INFö auf die Datei öDEKTOP.INFö auf Laufwerk öA:ö des Rechners mit der Bezeichnung USER zugreifen.

Zugriffsschutz

Auf mehrfachen Wunsch wurde ein Zugriffsschutz implementiert. Mit einem Markierungsfile MIDI_COM.LOC kann man nun eine Ordner Ebene gegen alle Zugriffe von öAußenö sperren. Die GEMDOS-Funktionen FCREATE, FOPEN, FDELETE und PEXEC werden abgebrochen, wenn Sie auf ein geschütztes File angewendet werden.

Diese Beschränkung bezieht sich dabei nur auf Files einer Ebene. Ordner in einer geschützten Ebene und deren Inhalte können, falls nicht ebenfalls geschützt, weiterhin von außen bearbeitet werden. Um nicht jeden Ordner einer Partition einzeln markieren zu müssen, wurde zusätzlich die Möglichkeit geschaffen, mit einer einzigen Markierung eine ganze Partition dem externen Zugriff zu entziehen. Man aktiviert alle beschriebenen Einschränkungen für eine ganze Partition durch das File M_C_LOCK.PAR im Rootdirektory. Beide Markierungsfiles haben die Länge ö0ö. Allein die Existenz eines Sperrfiles ist ausreichend für die Zugriffsbeschränkung.

NEU: Mittels der Status-Bits der Sperrfiles kann man nun auch eine READ-ONLY Bedingung erzeugen. Setzen Sie den Status auf NUR LESEN, so sind die Dateien der jeweiligen Partition bzw. Ordner Ebene nicht mehr gegen Lesen der anderen Netzteilnehmer geschützt sondern nur noch gegen Änderungen über das Netzwerk.

SICHERHEIT

Das Wort SICHERHEIT ist hier nicht umsonst groß geschrieben. Der Ring ist so konzipiert, daß Abstürze, Bootvorgänge oder herausgezogene Kabel nicht zu Datenverlusten führen (es sei denn, der sendende Rechner verabschiedet sich). Angefangene Operationen werden, sobald der Ring wieder geschlossen ist, zu Ende geführt. Dies bedeutet für Anwender, daß sie bei einer Unterbrechung des Rings auf die Wiederherstellung der Verbindung warten müssen. Es wurde kein Timeout vorgesehen.

Auf vielfachen Wunsch sperrt MIDI_COM nun aber das Netzlaufwerk, wenn der Verdacht auf eine Ringunterbrechung besteht. In den vorhergehenden Versionen bestand das Problem, daß ein Zugriff auf das Netzlaufwerk in einer Endlosschleife hängen blieb, wenn ein Netzteilnehmer inzwischen seinen Rechner abgeschaltet hatte. Nun wird nach etwa 3 Sekunden Unterbrechung kein Zugriff auf das Netzlaufwerk mehr zugelassen. Ein Zugriff innerhalb dieser Zeitspanne löst sich nur durch einen Neustart des eigenen Rechners oder des blockierenden (ausgeschalteten) Rechners zu Ende führen.

Intern überprüft MIDI_COM die Korrektheit der Daten mittels einer 32-Bit CRC-Prüfsumme, sodaß auch Übertragungsfehler praktisch ausgeschlossen werden können.

Bei Datei-Zugriffen über den Ring geht MIDI_COM in folgenden Schritten vor:

- a) Das zu lesende/schreibende File wird geöffnet.
- b) Der Lese-bzw. Schreibzeiger wird auf die gewünschte Position gesetzt.
- c) Die Lese-bzw. Schreiboperation wird ausgeführt.
- d) Das File wird wieder geschlossen.
- e) Das Ergebnis der Operation wird an den Auftraggeber übertragen.

Da zwischen Öffnen und Schließen der Datei aus Sicherheitsgründen keine Prozeßumschaltungen zugelassen werden (dies gilt auch für Multi-Tasking Erweiterungen), wird der Benutzer an dem entsprechenden Rechner zwangsläufig etwas behindert. Dafür ist aber im Normalfall (siehe Einschränkungen) gewährleistet, daß es nicht zu einer Zerstörung des Datenträgers kommen kann.

Da die interne Paketlänge einer Übertragung auf 4096 Bytes (normal 4 Sektoren) beschränkt ist, werden größere Lese- und Schreiboperationen durch MIDI_COM automatisch gestückelt. Dadurch werden bei größeren Operationen die zeitintensiven Schritte a)-e) mehrfach ausgeführt.

Die Druckumlenkung

Wie bereits angesprochen, lenkt MIDI_COM Druckerausgaben auf Drucker an externen Rechnern um, so daß ein Drucker im Ring durch alle angeschlossenen Rechner benutzt werden kann. MIDI_COM erlaubt auch mehrere Drucker im System, mit der Einschränkung, daß ein Rechner immer nur einen einzigen Drucker belegen kann. Jedem Rechner stehen alle Drucker im System zur Verfügung.

Die Auswahl eines Druckers kann entweder über das CPX-Modul oder mit Hilfe von MC_TALK.ACC oder MC_CONTR beeinflußt werden. Alle Ausgaben über die Standardkanäle 3 und -3 sowie alle Aufrufe der im Folgenden aufgeführten Betriebssystemaufrufe werden durch MIDI_COM umgelenkt.

BIOS 3 :	bconout
BIOS 8 :	bcostat
GEMDOS 5 :	cprnout
GEMDOS 17 :	cprnos

MIDI_COM geht dabei folgendermaßen vor:

Wird von einem Programm eine der oben beschriebenen Funktionen verwendet, so versucht MIDI_COM als erstes einen Drucker im System für den Rechner zu reservieren. Schließlich soll es ja nicht zu Überschneidungen bei der Ausgabe kommen. Gesucht wird dabei ein Drucker der ONLINE ist. Dabei wird die Suche auf die vorher getroffene Auswahl beschränkt. Bevorzugt wird immer der nächstliegende freie und empfangsbereite Drucker im Ring.

Wird ein Drucker reserviert, so wird eine 7 (BELL) an diesen übertragen. (Die meisten Drucker geben dann einen Pfeifton von sich). Der Drucker wird nach einer gewissen Zeit von MIDI_COM wieder freigegeben, wenn keine Druckzeichen mehr folgen. Dazu ist ein zweistufiger Timeout vorgesehen.

Wurde als letztes Zeichen ein Seitenvorschub übertragen, so wird eine andere Zeitspanne (im allgemeinen eine kürzere) gewählt, nach der der Drucker wieder für alle freigegeben wird, als bei einem normalen Timeout. Allerdings bleibt ein Drucker mindestens so lange belegt, bis der Spooler-Puffer leer ist. Über MC_CONTR können diese Werte eingestellt und permanent gesichert werden.

Zusätzlich kann vereinbart werden, ob bei einem normalen Timeout ein Seitenvorschub erzeugt werden soll. Dies soll es ermöglichen, daß der Drucker nach einem Benutzerwechsel wieder auf einen Seitenanfang druckt. Beachten Sie hierzu auch die Beschreibung von MC_CONTR.

Um nun den druckenden Rechner nicht zu stark zu belasten, ist in MIDI_COM zusätzlich ein Spooler eingebaut. Wird mit MC_CONTR ein Puffer vereinbart, so ist der Spooler automatisch aktiviert. Die Ausgabe aus dem Puffer wird durch einen Interrupt gesteuert, so daß im Normalfall der Anwender des ausdruckenden Rechners nicht behindert wird. Allerdings wird dabei ein ausreichend großer Puffer vorausgesetzt. Ein zu kleiner Puffer kann, falls er überläuft, zu kurzen Behinderungen am druckenden Rechner führen.

Utilities
MICO.CPX

Für Anwender des XCONTROL.ACC's von ATARI liegt ebenfalls ein CPX- Modul vor, mit dessen Hilfe die meisten der unter MC_CONTR beschriebenen Parameter geändert werden können. Allerdings ist es noch nicht möglich, diese Werte gleich abzuspeichern. Aber es eignet sich vorzüglich, die optimalen Werte für MIDI_COM herauszufinden. Nur der Wert von Timeout ist zur Zeit nicht über MC_CONTR voreinstellbar. Dieser Wert gibt an, nach etwa welcher Zeitspanne das Netz als unterbrochen gelten soll, also keine Zugriffe auf das Netzlaufwerk mehr möglich sind. Dieser Wert ist nur unter Verwendung der MIDI-Treiber von Belang.

Der Zweite CPX-Dialog wird über Drucker-Parameter geöffnet. Zusätzlich zu den Einstellmöglichkeiten bezüglich des Druckerverhaltens kann man von hier aus eine Synchronisierung der Uhrzeit und des Datums im Netz erzwingen. Durch einen Mausklick auf das Uhr-Symbol wird die aktuelle Uhrzeit und das Datum auf alle Rechner übertragen und dort ebenfalls gesetzt. Anders als unter MC_CONTR(*FN*s. Grundparameter Systemzeit setzen*FN*), wird diese Aktion sofort durchgeführt und nicht nur beim Booten eines markierten Rechners.

MC_STOP/MC_START

Auf mehrfachen Wunsch wurde MIDI_COM um die Möglichkeit erweitert, den MIDI-Ring abschalten zu können. Solange MIDI_COM aktiv ist, kreist ständig ein TOKEN über den Ring. Dies belastet den Rechner im allgemeinen nicht merklich, außer bei einigen speziellen, zeitkritischen Anwendungen, wie etwa der Betrieb eines HS-Modems. In diesem Fall kann man mit MC_STOP.PRG das Netzwerk stoppen und später mit MC_START.PRG wieder aktivieren.

Allerdings aktiviert sich MIDI_COM automatisch, wenn irgend ein Rechner im Netz neu gebootet wird.

VORSICHT:

Bei Aufruf von MC_STOP.PRG werden ALLE !! laufenden Netzoperationen kommentarlos abgebrochen. Daher sollte diese Option nur mit der nötigen Umsicht verwendet werden.

MC_WATCH

Dieses Programm stellt einen Überblick über die angemeldeten Applikationen im Netz dar. Bei der Entwicklung kann man mit diesem Programm die An- und Abmeldeaktivitäten im Netz kontrollieren. Die Source für dieses kleine Programm finden Sie auf der Originaldiskette. Wenn nicht, wenden Sie sich bitte an Ihren Händler. Dieser wird Ihnen gerne mitteilen, wie Sie mit dem Autor in Kontakt treten können.

Einschränkungen

1. Da MIDI_COM nur GEMDOS-Aufrufe abfängt und bearbeitet, sind einige Anwenderprogramme nicht in der Lage, auf das Netzlaufwerk zuzugreifen. Dies betrifft insbesondere die meisten Kopierprogramme, da diese mit Hilfe von BIOS oder XBIOS Aufrufen direkt auf die Sektoren der Laufwerke zugreifen. (Als Ausnahme ist das Kopierprogramm KOBOLD zu nennen, da dieses auf GEMDOS-Modus umschaltbar ist)
2. Trotz aller Maßnahmen kann es unter bestimmten Umständen zu Problemen mit dem Datenträger kommen. Prinzipiell sollte man sich vergewissern, das während des Einsatzes von Programmen, die direkt auf die Platte oder Diskette schreiben, also Schnelkopierer oder auch Diskmonitore etc, nicht über den Ring auf diese Partition geschrieben wird.
3. Hält ein Programm eine Datei geöffnet, und über MIDI_COM wird eine Datei gleichen Namens mittels FCREATE erzeugt, so legt das Betriebssystem eine zweite Datei mit dem gleichen Namen an. Das ist zwar unschön, gefährdet aber nicht den Datenträger.
4. Die XBIOS-Funktion PRINT-BLOCK wird nicht durch den Ring bedient. Ausserdem ist es nicht möglich mit ALT-HELP den Bildschirm zu drucken, wenn kein Drucker direkt angeschlossen ist.
5. Einige wenige TOS-Programme blockieren noch immer den Ring. MIDI_COM verwendet sowohl AES-Umschaltungen als auch die BIOS-Routinen zur Tastaturabfrage um sich Prozessorzeit zu verschaffen. Bis auf wenige Ausnahmen können daher auch öhne GEMDOS Programme verwendet werden.
6. Es können keine Programme vom Netzlaufwerk gestartet werden, die resident im Speicher verbleiben sollen, da MIDI_COM nach Programmende den Speicher immer freigibt. (Solche Programme muß man also erst auf einen eigenen Datenträger kopieren).
7. Es ist nicht möglich, die Anzahl der Rechner im Ring zu verändern nachdem sich der Ring einmal installiert hat. Im Eventualfall müssen daher alle Rechner ausgeschaltet werden und anschließend der Ring mit der neuen Rechnerzahl wieder installiert werden.
8. Speziell TEMPUS-Word kann leider nicht die Druckumlenkung verwenden, obwohl hier zwar die Option besteht, über das Betriebssystem zu drucken. Leider werden diese Aufrufe aber aus einem Interrupt heraus vorgenommen, was unter MIDI_COM zu Problemen führt.

Hilfe

Einige Probleme tauchten im Laufe der Zeit häufiger auf, sodaß diese hier kurz angesprochen werden sollen. Sollten trotzdem noch Fragen offen bleiben, wenden Sie sich bitte an Ihren Händler.

Laser-Drucker

Setzen Sie mcspooler: 0, so gibt MIDI_COM über normale Betriebssystem-Funktionen aus. In den meisten Fällen sollte der Laser-Drucker dann funktionieren. Der eingebaute Spooler gibt grundsätzlich direkt auf der normalen Druckerschnittstelle aus.

Drucken

Einige Textprogramme lassen sich beim Drucken zwischen den einzelnen Zeichen etwas zuviel Zeit. Dadurch entsteht immer wieder die Situation, das MIDI_COM den Drucker freigibt und anschließend beim nächsten Zeichen wieder belegt. Dadurch entstehen insbesondere bei Graphik-Drucken gelegentlich Schmutzzeichen. Als Lösung sollte man in diesem Fall mit den Werten von timeoutxx: und timeoutFF: etwas experimentieren. Je nach Rechner dürften Werte zwischen 300 bis 600 das Problem lösen.

Programmeinbindung

Ursprünglich wurde MIDI_COM entwickelt, um Programme kontrolliert über die MIDI-Schnittstelle kommunizieren zu lassen. (Artikel in der cöt 8/9 91) Einige dieser Optionen sind auch in der 3.93 Version noch enthalten. Die Anwendung der programmspezifischen Optionen ist aber nicht abhängig von irgendeiner Programmiersprache. Sollten Sie eigene Anwendungen/Spiele programmieren wollen, stehe ich gerne zur Verfügung.

Das Telefon im Programm MC_TALK ist nur eine mögliche Anwendung der direkten Datenübermittlung zwischen Programmen. Weitere denkbare Anwendungen wären z.B.: die Möglichkeit, ferngesteuert Hilfsprogramme zu starten, die dann praktisch als Batch-Jobs im Hintergrund (auf einem anderen Rechner) laufen.

Die Kommunikation zwischen einer Applikation und MIDI_COM benötigt natürlich ein definiertes Protokoll. Zum einen muß geklärt werden, wie MIDI_COM Daten übermittelt bekommt und zum anderen, wie diese Daten auszusehen haben.

Im folgenden finden Sie eine Beschreibung der Funktionen und die benötigten Parameter. Falls Sie bereits für frühere MIDI_COM-Versionen Programme geschrieben haben, finden Sie ausserdem ein Vergleich der Funktionen und Parameter, die MIDI_COM bis zur Version 3.9 verwendete.

Grundlagen

Die Kommunikation erfolgt über die Struktur `msg_typ(*FN*` s. Struktur `msg_typ*FN*)`. In älteren Versionen wurden noch die AES-Funktionen `APP_FIND/APP_WRITE` verwendet. Ab Version 3.7 benutzt MIDI_COM nun die selbstdefinierte GEMDOS-Funktion 111. Dadurch ist es nun auch mit TOS-Applikationen möglich, die Programmanbindung zu nutzen. Ist MIDI_COM nicht installiert, liefert der Funktionsaufruf einen negativen Fehlerwert, da diese Funktion im GEMDOS normalerweise nicht existiert.

Übergeben wird die Anfangsadresse des MSG-Puffers (früher der ganze Puffer) über den Stack. Die Struktur des MSG-Puffers hat sich nicht geändert, wohl aber etwas die Verwendung der Parameter. So sollte im letzten Eintrag (`msg.fill`) nun immer die durch ANMELDUNG zurückgelieferte ID eingetragen werden, damit MIDI_COM den Auftraggeber eines Befehls identifizieren kann. Bislang war dieser Eintrag unbenutzt.

Für PURE-C und HM2 finden Sie auf der beiliegenden Diskette bereits vorgefertigte Bibliotheken. Für andere Programmiersprachen müssen Sie sich die passenden Aufrufe noch selbst erstellen. Ausser unter Multi-TOS ist die Zentrale Funktion GEMDOS 111. Die Parameterübergabe unter Multi-TOS gestaltet sich etwas anders, da sich MIDI_COM in dieser Umgebung nicht in das Betriebssystem einbindet.

Vereinbarungen

Zeichenketten (z.B.: `önameö`) sind definiert als C-Strings. Sie beginnen direkt beim ersten Byte und enden mit einem 0-Byte. Eine leere Zeichenkette enthält an erster Stelle ein 0-Byte. In älteren MIDI_COM-Versionen wurden noch PASCAL-Strings verwendet. Diese enthalten als erstes Byte die Länge der Zeichenkette. Die Variable `mc_msg` ist im Folgenden vom Typ `msg_typ(*FN*` s. Strukturen `msg_typ*FN*)`.

Der Inhalt der einzelnen Felder ist abhängig vom Inhalt des Feldes `mc_msg.what`.

Anmelden

Mit diesem Befehl meldet sich eine Applikation bei MIDI_COM an. Die Applikation ist dafür verantwortlich, das mit der Anmeldung ein gültiger Empfangspuffer übergeben wird, in den MIDI_COM eingehende Daten schreiben kann. Der Name zur Anmeldung ist mit 8 Zeichen frei wählbar.

Parameter: `mc_msg.what` = `_ASSIGN;` `/* = 6000 */`

```
mc_msg.size    =    0;
mc_msg.ptr     =    io_ptr; /* adresse der io-Struktur */
mc_msg.name    =    name;  /* zeiger auf Appl-Namen  */
mc_msg.an      =    0;
mc_msg.fill    =    0;
```

RETURN: eindeutige ID der Applikation (kann von anderen Applikationen zur Adressierung verwendet werden) Oder eine Zahl <=0 wenn Anmeldung nicht möglich.

Abmelden

Die Applikation ist nicht länger empfangsbereit. Dieser Befehl sollte auf jeden Fall vor Programmende ausgeführt werden, da MIDI_COM sonst den Speicherbereich des Empfangspuffers weiterhin füllt, obwohl dieser inzwischen schon freigegeben wurde.

```
Parameter:      mc_msg.what   =      _GOOD_BYE; /* = 6001 */
                mc_msg.size   =      0;
                mc_msg.ptr    =      NULL;
                mc_msg.name   =      NULL;
                mc_msg.an     =      0;
                mc_msg.fill   =      apl_num;      /* Welche Appl wird abgemeldet */
```

Ringkarte anfordern

Durch diesen Befehl wird MIDI_COM aufgefordert, die Verwaltungsstruktur(*FN* s. Strukturen verwaltung*FN*) in den Empfangspuffer zu kopieren.

```
Parameter      mc_msg.what   =      _GET_VERW; /* = 6009 */

      mc_msg.size   =      0;

      mc_msg.ptr    =      NULL;

      mc_msg.name   =      NULL;

      mc_msg.an     =      0;

      mc_msg.fill   =      apl_num;      /* Appl_ID */
```

Der erste Eintrag ist der eigene Rechner, der zweite Eintrag die relative Adresse 1 und so weiter. Es werden grundsätzlich 7 Sätze übertragen.

Statusbits

Die Statusbits(*FN* s. Strukturen status*FN*) in der Verwaltungsstruktur haben kaum noch Bedeutung. Insbesondere die TOKEN-Bits sind praktisch nicht mehr existent, da MIDI_COM nun keine direkte Kontrolle mehr über den TOKEN ausübt. Diese Kontrolle wurde in den Treiber ausgelagert. Spätere Versionen werden eventuell ganz auf den Token verzichten. Dies ist bereits bei der ROM-Port-Version der Fall, die direkt mit Handshake arbeitet. Interessant bleiben für den Programmierer die unteren 3 Bits. Wenn alle drei Bits auf 0 ist der Status einem aktiven Rechner zugeordnet. Alle anderen Bits sind, insbesondere bei Stati zu externen Rechnern, Kopien des Zustandes zum Bootzeitpunkt und daher nur von geringer Aussagekraft.

Daten senden

Senden eines Datenpaketes an eine oder mehrere Applikationen. Die Paketlänge darf nicht größer als 4000 Bytes sein. Die Möglichkeiten zur Adressierung sind weiter unten beschrieben.

```
      mc_msg.what   =      _SEND_DATA;

      mc_msg.size   =      size;      /*Länge der Daten <4000 Bytes) */

      mc_msg.ptr    =      daten;     /* Zeiger auf Daten */

      mc_msg.name   =      name;      /* Zeiger auf einen Appl-Namen */

      mc_msg.an     =      Adresse;

      mc_msg.fill   =      apl_num;      /* Absender */

RETURN      1      =>  Übertragung erfolgt
```

- 1 => Übertragung TIME-OUT

Adressierung

Die Möglichkeiten zur Adressierung der Empfänger-Applikation(en) haben sich ebenfalls etwas geändert. Bedingt durch die Erweiterung auf 6 Programme pro Rechner, muß die Adressierung der Nachrichten zwangsläufig flexibler werden. Einen herzlichen Dank an A. Mayer, der mir dabei mit wertvollen Anregungen und Vorschlägen geholfen hat.

Ein Programm kann einen Datensatz an eine Variation verschiedener Programme auf verschiedenen Rechnern absenden.

1. Direkte Adressierung einer Applikation im Netz

mc_msg.an = apl_num einer anderen Applikation (Adressierung über ID)

mc_msg.name = Leerstring <=> 1. Byte ist eine 0

Früher: Nur Rechnernummer angegeben, Name leer

2. Senden an alle Programme im Netz, die sich unter dem adressierten Namen bei MIDI_COM angemeldet haben.

mc_msg.an = -1

mc_msg.name = 'xxxxx' (Name, unter dem sich eine oder mehrere Applikationen bei MIDI_COM angemeldet haben.

Früher: Name gesetzt, dann wurde an alle mit dem gleichen Namen verschickt

3. Adressierung eines Programms über den Namen auf einem anderen Rechner

mc_msg.an = relative Rechnernummer ($0 < \text{msg.an} < 7$)

mc_msg.name = 'xxxxx' (Name, unter dem sich die Applikation bei MIDI_COM angemeldet hat.

4. Adressierung aller Programme auf einem anderen Rechner.

mc_msg.an = relative Rechnernummer $0 < \text{msg.an} < 7$

mc_msg.name = Leerstring <=> 1. Byte ist eine 0

5. Sendung an ALLE

mc_msg.an = -1

mc_msg.name = Leerstring (1. Byte ist eine 0)

Daten Empfangen

Bei der Anmeldung wird die Adresse der Struktur `io_rec(*FN* s. Strukturen io_rec*FN*)` an `MIDI_COM` übergeben. Alle ankommenden Nachrichten werden von `MIDI_COM` mit Hilfe dieser Struktur abgelegt. Es liegt in der Aufgabe der Applikation diesen Puffer regelmäßig auszulesen.

Bemerkung zur Schreibweise:

```
help.aschar[0]=io_ptr.ibuf[io_ptr.nl++];
```

ist identisch mit

```
help.aschar[0]=io_ptr.ibuf[io_ptr.nl];  
io_ptr.nl=io_ptr.nl+1;
```

Mit Hilfe der folgenden Struktur können 2-Byte als Integer-Zahl oder als 2 Zeichen verwendet werden

typedef union

```
{  
    int    asint;  
    char   aschar[2];  
}konv1;
```

Programmbeispiel

Anhand des folgenden Programmbeispiels wird das Auslesen des Puffers beschrieben:

```
int lese(char *bufa1,int *esc,int *len)
{
    int    i;

    konv1  help;

    *esc=0;

    /* Die ersten zwei Bytes sind entweder die Länge der Nachricht oder eine ESC-Kennung.
    Zur Zeit gibt es davon nur eine einzige: _GET_VERW */

    /* Lese-Index = Schreibindex => Puffer LEER */
    if (io_ptr.nl==io_ptr.nw)    return(FALSE); /* Fertig */

    /* kopiere 1. Byte */
    help.aschar[0]=io_ptr.ibuf[io_ptr.nl++];

    /* Ist das Puffer-Ende erreicht, geht's wieder bei 0 weiter */
    if (io_ptr.nl==io_ptr.size) io_ptr.nl=0;

    /* kopiere 2. Byte */
    help.aschar[1]=io_ptr.ibuf[io_ptr.nl++];

    /* Ist das Puffer-Ende erreicht, geht's wieder bei 0 weiter */
    if (io_ptr.nl==io_ptr.size) io_ptr.nl=0;
    if (help.asint>4000)
    {

        /* Falls ESC-Kennung, dann kommt jetzt erst die Länge der Nachricht */

        *esc=help.asint;
        help.aschar[0]=io_ptr.ibuf[io_ptr.nl++];
        if (io_ptr.nl==io_ptr.size) io_ptr.nl=0;
        help.aschar[1]=io_ptr.ibuf[io_ptr.nl++];
        if (io_ptr.nl==io_ptr.size) io_ptr.nl=0;
        *len=help.asint-2;

    }

    /* Nun den eigentlichen Inhalt der Nachricht kopieren Die Länge enthält auch die beiden
    Bytes für die Längenangabe daher (-2) */

    for (i=0;i<help.asint-2;i++)
    {
        bufa1[i]=io_ptr.ibuf[io_ptr.nl++];
        if (io_ptr.nl==io_ptr.size) io_ptr.nl=0;
    }

    return(TRUE);
}
```

Das Treiber-Konzept

Um MIDI_COM in Zukunft auch mit anderen Schnittstellen als MIDI einsetzen zu können, wurden die Netzwerkfunktionen von den eigentlichen Schnittstellen-Funktionen getrennt. Der Treiber installiert nun einen COOKIE (MCTR), in dem er dann einige wenige Funktionen dem Netzwerk anbietet. Der Cookie-Wert zeigt auf eine Struktur(*FN* s. Strukturen io_port*FN*) mit den entsprechenden Einsprungadressen.

Auf der Diskette befinden sich die Source-Texte für den MIDI-Treiber. Wer also Interesse hat, eigene Treiber für die reichhaltigen Schnittstellen der ATARI-Computer zu schreiben, kann mit meiner vollen Unterstützung rechnen.

Treiberfunktionen

Die Hauptfunktionen sind natürlich lesen und schreiben von bzw. auf der Schnittstelle. Alle Funktionen werden im Supervisormodus aufgerufen Die Funktion

```
int (*LESE)(uint *len,char *buff);
```

wird durch MIDI_COM regelmäßig aufgerufen. Diese liefert die Länge eines empfangenen Datensatzes und einen Zeiger auf das erste Byte der Daten im Speicher.

Als Return-Wert erwartet MIDI_COM ein Bitmuster

```
typedef struct {
    int    frei:8; /*    freie Bits    */
    int    erg:2;  /*    Ergebniss TRUE/FALSE/FAIL */
    int    doubl:1;/*    Duplikat    */
    int    reply:1;/*    REPLY-Kennung    */
    int    von:4;  /*    Adresse Absender    */
}r_wert;
```

Das Ergebniss ist TRUE (erg = '01'), wenn eine Nachricht im Empfangspuffer zur Bearbeitung vorliegt. Andernfalls wird FALSE (erg = '00') zurückgeliefert. Der Wert FAIL (erg = '11') wird erwartet, wenn die Schnittstelle nicht zur Verfügung steht. Dies ist z.B. der Fall, wenn der MIDI_RING unterbrochen ist. Der MIDI-Treiber antwortet nach einer gewissen Zeit mit FAIL, sodaß MIDI_COM das Netzlaufwerk sperren kann.

Die Duplikaterkennung wird von MIDI_COM zur Zeit noch nicht ausgewertet.

Die Reply-Kennung ist ö1ö, wenn es sich bei der Nachricht um eine Antwort auf eine öAuftragssendungö handelt.

Ausserdem enthält der Rückmeldewert noch die Nummer des Absenders, also einen Wert zwischen 0 und 7.

Als weitere Funktionen gibt es zwei Schreibfunktionen.

```
int      (*SENDER)(int an,uint LEN,char *buff);
```

```
int      (*REPLY)(uint LEN,char *buff);
```

Um einen Auftrag an ein anderes MIDI_COM abzusetzen, wird die Funktion SENDER aufgerufen. Dies enthält ausser der Nachrichtenlänge und einem Zeiger auf die Nachricht zusätzlich die Adresse des oder der Empfänger der Nachricht als Bitmuster.

Die REPLY-Funktion wird (wenn überhaupt) direkt nach dem Lesen einer Nachricht aufgerufen. Der Treiber hat in diesem Fall die Aufgabe, sich jeweils die Absender-Adresse der zuletzt gelesenen Nachricht zu merken. Nicht alle Nachrichten werden jedoch beantwortet. Insbesondere wenn es sich bereits um eine Antwort handelt.

Als vierte Funktion ist eine allgemeine Steuerfunktion vorgesehen. Über diese erfragt MIDI_COM unter anderem seine interne Nummer für die Adressierungen.

```
int      (*CMD)(int cmd,int *par1,int *par2);
```

Es existieren zur Zeit die Kommandos:

anmeldung MIDI_COM gibt dem Treiber Zeit sich zu installieren. Eventuelle Einbindungen in Interrupts oder Betriebssystemvariablen sollten bei diesem Aufruf vorgenommen werden.

new_appl Wird anmeldung aufgerufen, liefert diese Funktion auf allen anderen Rechnern für einen Aufruf den Wert TRUE. Dies ist die Kennung, ob sich in der Netztopologie etwas geändert hat.

err_bell Die Fehlerkontrolle beziehungsweise die Anzeige der Fehler an der Schnittstelle wird eingestellt. Über par1 wird ein Wert 0,1 oder 2 mitgegeben.

id_request Abfrage der eigenen ID des Treibers. der Wert sollte zwischen 0 und 6 liegen.

stop Bei der MIDI-Schnittstelle löst der Treiber zur Kontrolle einen TOKEN Kreisen. Mit diesem Kommando wird der Betrieb eingestellt und die Schnittstelle gesperrt.

start Gegenbefehl zu stop

Strukturen

typedef struct

```
{  
    int    what; /* 2 Byte */  
    int    size; /* 2 Byte */  
    char   *ptr; /* 4 Byte => Adresse des ersten Zeichens */  
    char   *name; /* 4 Byte => Adresse des ersten Zeichens */  
    int    an; /* 2 Byte */  
    int    fill; /* 2 Byte */
```

} msg_typ;

typedef struct

```
{  
    char    my_name[6][10]; /* 6 Appl. Namen (a 10 Byte) */  
    char    r_name[10]; /* Rechnername im Direktorie */  
    bit_test status; /* integer; { Status-Bits } */  
    int     disk_st; /* integer; */
```

} verwaltung;

typedef struct

```
{  
    char    *ibuf; /* Zeiger auf den buffer */  
    int     size; /* gr"o"e des Buffers */  
    int     nw; /* n"ochste Leseposition */  
    int     nl; /* n"ochste Schreibposition */  
    int     blow; /* nicht verwendet */  
    int     bhig; /* nicht verwendet */
```

} io_rec;

```

typedef struct
{
    unsigned    rest:4;
    unsigned    stat:1; /*bit 11: Statussendung bekommen */
    unsigned    trns:1; /*bit 10: Druck file_transfer on */
    unsigned    prfree:1; /*bit 9 : 0 = Drucker frei */
    unsigned    appl:1; /*bit 8 : habe Appl. */
    unsigned    prnt:1; /*bit 7 : habe einen Drucker */
    unsigned    rts:1; /*bit 6 : ready to send */
    unsigned    prf:1; /*bit 5 : prfile in arbeit */
    unsigned    wt:1; /*bit 4 : warte auf token */
    unsigned    token:1; /*bit 3 : have token */
    unsigned    rtw:3; /* ready to work */
} status;

/* zu Status: bit 0 : sync erhalten */
/* bit 1 : test erhalten */
/* bit 2 : ready to work */
/* Bit 5 ist gesetzt, wenn kein eigener Drucker vorhanden ist, und der Text */
/* übertragen werden muß */

```

```

typedef struct {
    int      (*SENDER)(int an,uint LEN,char *buff);
    int      (*REPLY)(uint LEN,char *buff);
    int      (*LESE)(uint *len,char *buff);
    int      (*CMD)(int cmd,int *par1,int *par2);
    x_buffer    *xbuffer;
}io_port;

```