

MIDI/RS232-80C535
Handbuch zur Hardware
Stand 01.10.1999

Wichtiger Hinweis:

Sowohl die vorliegende Software, als auch die dazugehörige Hardware, wurden mit größter Sorgfalt erstellt. Trotzdem lassen sich Fehler nie mit Sicherheit ausschließen. Daher können wir keinerlei Haftung für irgendwelche Schäden übernehmen, die durch den Gebrauch dieser Software oder Hardware entstehen könnten.

Warennamen werden in diesem Dokument ohne die Gewährleistung einer freien Verfügbarkeit verwendet. Viele der verwendeten Bezeichnungen sind gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden.

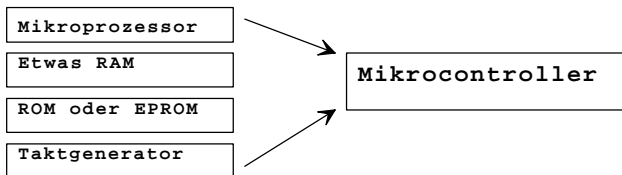
Dieses Handbuch und die dazugehörige Software sind urheberrechtlich geschützt. Die Weitergabe und/oder Vervielfältigung ist nicht erlaubt!

**© 1993-99 Dipl.-Ing. Jürgen Wickenhäuser
Rastatter Str. 144
D-76199 Karlsruhe
Telefon: +49 (0) 721 / 9 88 49 - 0
Fax: +49 (0) 721 / 88 68 07
Internet: WWW.WICKENHAEUSER.DE**

2 Mikrocontroller

2.1 Eine Übersicht

Was ein Mikroprozessor ist, weiß inzwischen ja wohl jeder. Dagegen ist der Begriff Mikrocontroller den meisten weniger geläufig. Der Grund: Mikrocontroller sind 'Heinzelmännchen'; sie arbeiten fast immer im Verborgenen: in Videorecordern, Waschmaschinen, Aufzügen, in Spielzeug, Alarmanlagen und sogar in Bohrmaschinen! Mit Sicherheit gibt es heutzutage bedeutend mehr Mikrocontroller auf der Welt als "reine" Mikroprozessoren. Was aber ist der Unterschied: Zum Aufbau eines "klassischen" Computersystems werden mindestens die folgenden Komponenten benötigt:



Mikrocontroller

Ein Mikrocontroller ist die Summe dieser 5 Teile in einem einzigen IC. Daher werden Mikrocontroller oft auch als Ein-Chip-Computer bezeichnet, was den Sachverhalt sehr genau beschreibt: Die gesamte Peripherie eines Mikrocontrollers besteht im Minimalfall aus einer Batterie und einem Quarz.

Mit nur wenigen zusätzlichen Komponenten lassen sich mit Mikrocontrollern vielseitige Steuer-, Meß- und Schaltungsaufgaben wahrnehmen.

Besonders die "51-er" Familie ist weltweit sehr stark verbreitet, da sie über viele Mitglieder und ein breites Leistungs-Spektrum verfügt:

Name	ROM oder EPROM auf dem Chip selbst	Internes RAM auf dem Chip	Besonderheiten
AT89C2051	2kB Flash-E ² PROM	128 Bytes	20-poliges Gehäuse
DS87C520	16kB EPROM	256 Bytes + 1kB	Eff. Taktfrq. 99 MHz
8031	keines	128 Bytes	
8052	8kB ROM	256 Bytes	
87C51	4kB EPROM	128 Bytes	Die 'C' Typen brauchen nur wenig Strom
80C31	keines	128 Bytes	
C515	keines	256 Bytes	8 A/D-Wandler mit auf dem Chip
80C535	keines	256 Bytes	
...

Die "51-er"-Mikrocontroller-Familie umfaßt einige Dutzend Mitglieder

Einige der "51-er" besitzen keinen Programmspeicher auf dem Chip (wie etwa der 8031). Diese sind dann keine echten Ein-Chip-Computer, da sie zusätzlich ein ROM oder EPROM benötigen. Manche der Typen sind für kleinere Anwendungen

erst bei hohen Stückzahlen. Mikrocontroller mit EPROM (8751, 87C51, ...) lassen sich mit den entsprechenden handelsüblichen EPROM-Programmiergeräten programmieren. Das 'C' im Namen mancher Typen (etwa 80C31) bedeutet, daß es sich um einen CMOS-Typ handelt. CMOS-Typen brauchen nur sehr wenig Strom. Die Verwandtschaft der Familie besteht aus zwei Eigenschaften: Zum einen besitzen alle eine minimale gemeinsame Anzahl von Ports, Timern und RAM, **zum anderen sind die CPUs aller Typen identisch!** Das heißt: **Der Befehlssatz der CPUs ist 100% identisch.** Obwohl der Befehlssatz an sich recht einfach ist, ist er trotzdem auch effizient, und vor allem schnell: Für eine 8x8-Bit Multiplikation oder eine 16/8-Bit Division benötigen selbst die langsamsten der Familie nur 4 Mikrosekunden! Alle 51-er sind somit Software-aufwärtskompatibel zum 8031. Die zusätzlichen Funktionen der größeren Geschwister des 8031 (sie haben mehr Ports, mehr RAM, ROM oder EPROM, Timer; sie sind schneller; ...) lassen sich durch ein schlaues Softwarekonzept ohne Änderungen mit ein und dem selben Assembler nutzen. Doch davon später mehr...

2.2 Anwendung entwickelter Programme

Läuft Ihr entwickeltes Programm wunschgemäß, bestehen für Sie im wesentlichen 3 Möglichkeiten:

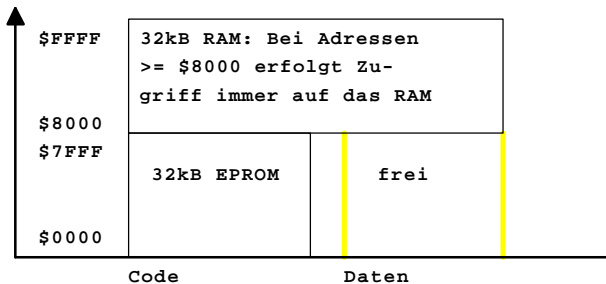
- 1.) Sie brennen das Programm in ein EPROM und ersetzen das Original-EPROM des MR durch Ihr eigenes.
- 2.) Sie verwenden Ihr EPROM für eine komplette Eigenentwicklung.
- 3.) Als dritte Alternative kann das Programm in einen Mikrocontroller mit eingebautem EPROM gebrannt werden, womit Sie sich ihren persönlichen **Ein-Chip-Computer** verwirklicht hätten, sofern Ihr Programm in das EPROM des entsprechenden Chips paßt.

Eine in diesem Zusammenhang besonders interessante Produktreihe stammt von der Firma ATMEL (www.atmel.com): Die beiden "Winzlinge" AT89C1051 und AT89C2051 befinden in einem nur 20-poligen Gehäuse, besitzen 1kB oder 2kB Flash-E²prom, es kann bis zu 1000 mal reprogrammiert werden! Vor allen Dingen sind diese echten "Mikros" auch sehr preiswert.

3 Die Hardware (Übersicht)

3.1 Systemkonfiguration der Hardware MR

Die Hardware wurde möglichst einfach gehalten. Den Schaltplan finden Sie im Anhang. Normalerweise können die Mitglieder der 51-er Familie nur Programme im EPROM ausführen, d. h. sie können mit ihren 16 Adress-Leitungen jeweils 64kB Programme und 64kB Daten adressieren, macht zusammen 128 kB. Um aber in der Entwicklungs- und Testphase Programme im RAM ausführen zu können, bedarf es eines kleinen Tricks. Mittels des 74HCT00 wird bei Speicherzugriffen über die Adresse \$8000 immer das RAM adressiert, egal, ob es sich um Code oder Daten handelt. Zwar reduziert sich dadurch der maximal mögliche Speicherbereich um 32kB, was aber in den seltensten Fällen ein Hindernis darstellen dürfte.



Speicheraufteilung

Im EPROM liegen die Kommunikationsroutinen. Das RAM ist vollständig für den Anwender verfügbar!

Will ein 51-er einen Befehlscode oder Daten aus dem Programmspeicher (Tabellen, Konstanten) holen, so gibt jener die Adresse aus und setzt danach das Signal PSEN- auf LOW. Bei Lesezugriffen auf den RAM-Speicher wird nach der Adressausgabe das Signal RD auf LOW gesetzt, bei Schreibzugriffen das Signal WR-. Normalerweise würden also bei gleicher Adresse verschiedene Speicherbausteine angesprochen!

Desweiteren gibt es im allgemeinen keine Möglichkeit, Programme im RAM eines 51-er-Systems ablaufen zu lassen. Durch den Schaltungskniff mit dem NAND-Gatter 74HCT00 werden beide Adressräume bei Zugriffen auf Adressen oberhalb von \$8000 vereinigt.

3.2 Schnittstellen

Eine echte Besonderheit von MR sind seine beiden möglichen Schnittstellen-Konfigurationen: Da ist zum einen die MIDI-Schnittstelle (MIDI-Schnittstellen sind in der Musikwelt sehr verbreitet, daher auch ihr Name: "Musical Instrument Data Interface"), und zum anderen die RS232-Schnittstelle. Der Vorteil der MIDI-Schnittstelle ist die galvanische Trennung zwischen MR und PC oder Synthesizer. Der Nachteil der MIDI-Schnittstelle ist, daß sie auf PCs nicht serienmäßig eingebaut ist.

TIP: Prinzipiell arbeiten MIDI-Schnittstellen mit einer Übertragungsrate von 31250 Baud. Wenn Sie allerdings nur an der galvanischen Trennung Ihrer MR-Hardware und Ihres PCs interessiert sind und weiterhin nur gewöhnliche RS232-Datenübertragungsraten verwenden möchten, können Sie mit einem weiteren Optokoppler und einem RS232-Pegelumsetzer (auf einer Zwischenplatine) eine vollständige galvanische Trennung erreichen.

MR überläßt Ihnen die Wahl der Konfiguration: Bei Einbau der im Bausatz enthaltenen beiden 5-poligen DIN-Buchsen, können Sie wahlweise die MIDI- oder RS232-Betriebsart verwenden, je nachdem welchen Treiber IC Sie in Ihrer Hardware momentan eingesetzt haben, da beide, MIDI-Signal und RS232-Signal, an der Buchse abgegriffen werden können. Bei Verwendung der 5-poligen DIN-Buchse in RS232-Betriebsart ist folgendes zu beachten: Während Sie bei der Wahl der MIDI-Betriebsart die MIDI-Kabel direkt einstecken können, müssen Sie sich bei der Wahl der Betriebsart RS232 selbst ein Kabel herstellen. Das ist aber prinzipiell kein Problem, da dieses Kabel nur 3 Adern enthält. Der Aufbau von Verbindungskabeln wird im Kapitel 4 ausführlich beschrieben. Bei Bedarf können Sie auch auf die MIDI-Fähigkeiten von MR verzichten, und anstelle der beiden 5-poligen DIN-Buchsen einen abgewinkelten, 9-poligen Sub-D-Stecker verwenden, wie er allgemein für RS232-Schnittstellen verwendet wird. Das Verbindungskabel zwischen MR-Hardware und PC kann dann aber als sogenanntes "Null-Modem-Kabel" fertig gekauft werden. Sie können es natürlich auch selbst herstellen, wie in einem nachfolgenden Kapitel erläutert wird. Wir haben dem Bausatz nur die 9 polige SUB-D-Buchse beigelegt.

4 Aufbau der Hardware MIDI/RS232-80C535

4.1 Buchsenleisten und Stecker...

Wie bereits im Kapitel 3.2 angeschnitten, haben Sie beim Aufbau der Hardware teilweise mehrere Möglichkeiten bezüglich der verwendeten Stecker bzw. Buchsen:

Die 40 frei verwendbaren Portpins des Mikrocontrollers müssen irgendwie beschaltet werden können. Auf der Platine zu MR gibt es dazu zwei Alternativen:

- je eine 20- und 32- oder 30-polige Doppel-Buchsenleiste
- je eine 20- und 34-polige DIN-Steckerwanne (DIN 41651)

Im Bausatz sind die Doppel-Buchsenleisten enthalten, da wir meinen daß bei einem Entwicklungs-System alle Portpins leicht zugänglich sein sollten. Sie können in die beiliegenden Buchsenleisten notfalls **direkt** abisolierte Kabelenden oder Bauteile einstecken, wogegen Sie beim Einsatz von Steckerwannen immer einen eigens angefertigten Stecker benötigen. In einem Seriengerät dagegen sind Steckerwannen sicher die sinnvollere Lösung.

Das zweite Problem ist die Verbindung der MR-Hardware mit ihrem PC. MR unterstützt 2 Schnittstellen-Typen: MIDI und RS232, je nach eingesetztem Treiber-IC. Der Vorteil von MIDI ist die galvanische Trennung der beiden Kommunikationspartner über Optokoppler und die standardmäßig hohe Datenübertragungsrate von 31250 Baud (das sind über 3kB pro Sekunde). Der Nachteil von MIDI ist, daß MIDI-Schnittstellen serienmäßig nicht auf PCs verfügbar sind. An den im Bausatz enthaltenen 5-poligen DIN-Buchsen lassen sich direkt MIDI-Kabel einstecken. Auf den beiden freien Anschlüssen der näher in der Platinenmitte gelegenen 5-poligen DIN-Buchse liegt bei Einsatz des RS232-Treiber-ICs das RS232-Signal an. Sie können daher an den 5-poligen DIN-Buchsen sowohl Kommunikation über MIDI, als auch über RS232 betreiben.

Bei der Verwendung der DIN-Buchsen in der RS232-Betriebsart müssen Sie sich die passenden Verbindungskabel zwischen der MR-Hardware und ihrem PC selbst herstellen, wie in Kapitel 4.6 beschrieben.

Kommt für Sie ein Betrieb der Schnittstelle nur im reinen RS232-Betrieb in Betracht, sollten Sie den 9-poligen Sub-D-Stecker verwenden. Der Einsatz eines Sub-D-Steckers hat den Vorteil, daß Sie als Verbindungskabel von ihrem Rechner zu MR handelsübliche, sogenannte "Nullmodem-Kabel" verwenden können, wie in Kapitel 4.6 beschrieben.

Ansonsten können Sie über die MIDI-Schnittstelle auch andere Datenübertragungsraten als die MIDI-Standards übertragen, sofern sich am anderen Ende der Leitung ein geeigneter Kommunikationspartner befindet.

4.2 Vorbereitungen vor dem Zusammenbau

Sie benötigen:

- Elektronik-LötKolben mit maximal 30 Watt
- Elektronik-Lötzinn
- Seitenschneider
- eine kleine Zange zum Einpressen der Lötnägel
- eventuell ein Meßgerät (Multimeter)

Zum Betrieb der Schaltung **MIDI/RS232-80C535** benötigen Sie eine Stromversorgung. Das kann eine einfache 9-Volt-Batterie sein, oder ein kleines Netzteil mit einer Spannung zwischen 7 und ca. 12 Volt.

In der Schaltung werden viele ICs in CMOS-Technologie verwendet. Früher war der Umgang mit CMOS-ICs nicht ganz unproblematisch, da sie gegenüber elektrostatischer Aufladung empfindlich waren. Heutzutage sind CMOS-ICs durch Schutzdioden, die mit auf dem Chip integriert sind, kaum empfindlicher als andere Bauteile. Trotzdem empfiehlt es sich, beim Zusammenbau der Schaltung die Gefahr starker elektrostatischer Aufladung möglichst zu vermeiden (zum Beispiel: direkter Kontakt der ICs mit Mattscheiben eingeschalteter Bildschirme, sofortiges Berühren der ICs nach Gang über neue Teppichböden) und die Teile erst bei Gebrauch aus ihrer Verpackung zu entnehmen.

Falls Sie Probleme oder Fragen beim Zusammenbau haben: rufen Sie uns ruhig an, wir helfen Ihnen gerne weiter.

4.3 Stückliste

Überprüfen sie bitte **vor** dem Zusammenbau, ob der Bausatz komplett ist. *Die kursiv aufgeführten Bauteile sind nur im optional zusätzlichen MIDI-Kit enthalten.*

ICs (antistatisch verpackt):

- IC1: Mikrocontroller 80C535 (nur in CMOS-Ausführung!) im 68-poligen PLCC-Gehäuse
- IC2: Latch 74HCT573, 20-polig
- IC3: gebranntes Entwicklungs-System EPROM 27C256, 28-polig (evtl mit Aufkleber: schützt ein evtl. vorhandenes Lösch-Fenster des EPROMs vor Licht und damit vor langfristigem Datenverlust.). Das

Eprom ist nur vorhanden, wenn Sie MR als Bundle zusammen mit Software gekauft haben (μ C/51-Vollversion oder ASSEMBLER/51). Bei Verwendung der μ C/51-Demoversion müssen Sie das Eprom separat bestellen oder selbst brennen!

- IC4: 32-kB RAM Typ 62256 oder 51256 oder 43256 oder Vergleichstyp, 28-polig
- IC5: *Treiber-IC für MIDI-Schnittstelle: Optokoppler PC900, 6-polig (im Bestückungsaufdruck nur mit I5 bezeichnet)*
- IC6: Gatter 70HCT00, 14-polig
- IC7: Spannungsregler 7805, TO220-Gehäuse
- IC8: MAX232 Treiber IC für RS232-Schnittstelle: Pegelumsetzer, 16-polig (evtl. ist dieser IC auch als ITS80272 bezeichnet)

Halbleiter:

- D1: Dioden 1N400x (dick, schwarz)
- D2: *Diode 1N4148 (kleiner Glaskörper)*
- LED1: Leuchtdiode (LED)

Widerstände:

- R1, R3,R4,R6: 150 Ohm (Ringe: braun/grün/braun)
- R2: 270 Ohm (rot/lila/braun)
- R5: *1 kOhm (braun/schwarz/rot)*

Kondensatoren:

- C1: 220 μ F, Elko axial
- C2: 47 μ F, Elko radial
- C3,C6, C7,C8: 100 nF (Aufschrift "104" oder ".1" oder ähnlich)
- C4,C5: 33 pF
- C9,C10,C11,C12,C13,C14: 10 μ F, Elko radial

Sonstiges:

- 1 IC-Sockel PLCC-68, 68-polig
- 2 IC-Sockel DIL28, 28-polig
- 1 IC-Sockel DIL20, 20-polig
- 1 IC-Sockel DIL16, 16-polig
- 1 IC-Sockel DIL14, 14-polig
- 1 *IC-Sockel DIL6, 6-polig*

- CON1: Doppel-Buchsenleiste 32-polig (diese kann auch aus einer einzelnen 32-poligen Einzel-Buchsenleiste bestehen, die sich leicht in gleichgroße Teile zerbrechen läßt, z.B. mit einem Seitenschneider. Sollten Sie beim Zerbrechen der Buchsenleiste diese beschädigen, können Sie von uns eine neue erhalten, wenn Sie uns einen frankierten Rückumschlag (Standardbrief) zusenden.)
- CON2: Doppel-Buchsenleiste 20-polig (oder eine 20-polige Einfach-Buchsenleiste, siehe CON1)
- Q1: Quarz 12.000 MHz
- T1: Mini-Taster
- CON4: 9-poliger Sub-D-Stecker *oder M1,M2: je eine DIN-Buchsen 5-polig*
- 2 Lötnägel (= CON3)
- 4 Schrauben M3 x 16 mm
- 4 Abstandshülsen
- 4 Muttern M3
- 1 Platine **MIDI/RS232-80C535** mit Bestückungsaufdruck

4.4 Zusammenbau

Das wichtigste Kriterium beim Entwurf der Platine war die möglichst hohe Nachbausicherheit. Auch wenn Sie möglicherweise nur wenig Löterfahrung besitzen, dürfte Ihnen der Aufbau von **MIDI/RS232-80C535** keinerlei Probleme bereiten. Bei der Platine handelt es sich um ein qualitativ hochwertiges Erzeugnis. Sie ist bereits verzinnt und mit Lötstop-Lack überzogen.

Stellen Sie vor dem Zusammenbau sicher, daß Sie Elektronik-Lötzinn verwenden! Andere Lötzinn-Arten enthalten teilweise elektrisch leitende oder korrodierende Zusätze!

Zuerst werden die 6 IC-Fassungen und die Buchsenleisten eingelötet. Achten Sie darauf, daß die Markierungskerben der Sockel mit dem Bestückungsaufdruck übereinstimmen! Beim PLCC-68-Sockel muß die angeschrägte Seite mit dem Bestückungsaufdruck übereinstimmen.

Als nächstes folgen die restlichen Bauteile, inklusive IC7. Die IC-Sockel bleiben momentan noch leer. Bei den 2 Dioden und der LED ist auf die Polarität zu achten: Der Strich auf der Diode ist auf der gleichen Seite wie der Strich auf dem Bestückungsaufdruck (Eine Abbildung dazu finden Sie im Anhang dieses Handbuchs). Bei der LED wird der kürzere der beiden Anschlüsse auf der Seite mit dem Punkt eingelötet. Auch bei den Elkos muß die aufgedruckte Polarität auf dem Bauteil mit dem Bestückungsaufdruck übereinstimmen (Bei Elkos ist normalerweise

die Minus-Seite durch einen aufgedruckten Pfeil gekennzeichnet). Die Lötnägel werden mit der Zange unter leichtem Drehen in die Löcher gedrückt und großzügig auf beiden Seiten der Platine verlötet.

Überprüfen sie nach Einbau aller Bauteile die Platine noch einmal auf Kurzschlüsse und nicht verlötete Pins!

Probekalber kann nun Spannung an die Platine angelegt werden (noch ist keiner der gesockelten ICs eingesetzt). Falls sie ein Meßgerät zur Hand haben, können sie nun die stabilisierte 5-Volt Versorgungsspannung kontrollieren. Die zulässige Toleranz der 5-Volt Versorgungsspannung beträgt +/- 0.5 Volt. Schalten sie die Spannung wieder ab.

Zum Schluß werden die ICs eingesetzt. Achten sie auf die Übereinstimmung der Kerben an Bauteil und Sockel! Bei einigen ICs ist anstelle einer Kerbe ein Punkt aufgedruckt (zum Beispiel beim Optokoppler und beim Mikrocontroller). Der Punkt bezeichnet dann den Pin1 (eine Tabelle der verwendeten ICs finden sie im Anhang des Handbuches).

Von den beiden ICs IC5 (Optokoppler) und IC8 (MAX232) dürfen Sie nur einen von beiden einsetzen, je nach gewünschter Betriebsart der seriellen Schnittstelle!

Die Beinchen fabrikneuer ICs stehen gewöhnlich etwas ab. Das Ausrichten einer Beinchenreihe eines ICs geht am besten, indem Sie ihn seitlich, hochkant gegen eine glatte Fläche drücken (Tischplatte).

Wenn sie nun Spannung anlegen muß die LED auf der Platine zuerst mehrmals schnell, danach langsamer blinken. Genau das gleiche passiert bei Druck auf die Reset-Taste. Das Blinken der LED signalisiert, daß **MIDI/RS232-80C535** kommunikationsbereit ist.

TIP: Zum Entfernen des 80C535 aus seiner Fassung werden spezielle Werkzeuge angeboten. Wir meinen, daß Sie sich diese Ausgabe sparen können: Verwenden sie zum Entfernen des 80C535 2 dünne Kreuzschlitz-Schraubendreher, mit denen sie den 80C535 von beiden Seiten der Fassung aus gleichzeitig heraushebeln.

TIP: Das Auswechseln des EPROMS ist durch seine zentrale Lage auf der Platine etwas schwierig. Stecken Sie daher zwischen die eingelötete Fassung und das EPROM eine weitere 28-polige Fassung, wenn Sie das EPROM öfters wechseln möchten.

4.4.1 Fehlerbehebung

Falls Ihr **MIDI/RS232-80C535** nicht richtig funktionieren sollte: **keine Panik!**
Gehen sie folgendermaßen vor:

- Überprüfen sie zuerst, ob Sie auch wirklich alle Anschlüsse angelötet haben. Auf der verzinnten Platine werden vergessene Lötstellen leicht übersehen.
- Entfernen Sie alle ICs aus ihren Sockeln und legen Sie Spannung an. Die stabilisierte Versorgungsspannung muß zwischen 4.5 und 5.5 Volt liegen, die Stromaufnahme bei etwa 3 bis 6 Milliampere (nur IC7 verbraucht jetzt Strom).
- Schalten sie aus und setzen sie als nächstes IC1 (80C535), IC2 (74HCT573), IC3 (27C256) und IC6 (74HCT00) ein. Legen sie erneut Spannung an: Auch ohne IC4 (32kB-RAM) und IC5 (Optokoppler) und IC8 (RS232-Treiber) muß **MIDI/RS232-80C535** nun blinken. Die Stromaufnahme liegt bei etwa 30 Milliampere plus 20 mA für die LED. Sollte die LED nicht blinken, besteht der Verdacht, daß IC1 (80C535) oder IC3 (27C256) defekt sein könnten.
- Ein mechanisch etwas empfindliches Bauteil ist das Quarz: Durch mechanische Beanspruchung (z. B. Umbiegen des eingelöteten Quarzes) läßt sich das Quarz beschädigen. Falls sie im Besitz eines Oszilloskopes sind, können sie bei ordnungsgemäßer Funktion des Oszillators am Mikrocontroller-Pin 40 (XTAL1) eine Schwingung von 12MHz messen. Im Zweifelsfall können sie auch das mitgelieferte Quarz probelhalber durch einen beliebigen anderen Typ zwischen 1 und 12 MHz ersetzen. Allerdings arbeiten die im EPROM enthaltenen Kommunikationsroutinen nur mit einem 12MHz Quarz korrekt!
- Blinkt die LED können sie mit dem Programm 'FLASHMON' auf der CD die Datenübertragung zwischen MIDI/RS232-80C535 und ihrem PC überprüfen (Sie müssen dazu die Software vorher installieren): Bei gestartetem FLASHMON (9600 Bd) muß dieser einen Reset des MC mit dem Text "<external Reset>" anzeigen. Sie können nun auch noch probelhalber versuchen lassen ein Flash-E²prom zu identifizieren, FLASHMON lädt dazu einen kleinen Treiber ins RAM des MR. Dieser Treiber muß das "Flash" als "Unknown Device" identifizieren (es ist ja auch keines vorhanden).
- Bekommen sie ihre Hardware überhaupt nicht zum Laufen, sollten sie sich ruhig telefonisch an uns wenden, ebenso wenn Sie Ersatzteile benötigen!

4.5 Ersatzteile

Einige Bauteile von **MIDI/RS232-80C535** sind nur schwer zu bekommen. Wir bieten Ihnen daher an, die Ersatzteile zu einem fairen Preis über uns zu beziehen.

Bitte haben Sie Verständnis dafür, daß wir Bauteile nicht umtauschen können. Sämtliche Bauteile wurden beim Hersteller geprüft, es handelt sich nur um Ware 1. Wahl!

4.6 Herstellung eines Verbindungskabels

Beabsichtigen Sie, die serielle Schnittstelle von MR in der MIDI-Betriebsart zu verwenden und sind Sie bereits im Besitz von 2 MIDI-Kabeln, können sie den folgenden Unterabschnitt überspringen.

Die Verbindung des Rechners mit MR durch 2 MIDI-Kabel:

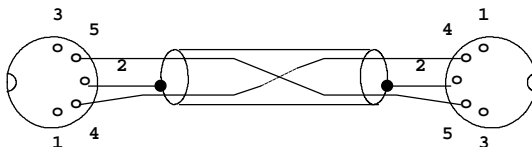
- 'MIDI-OUT' des Rechners wird in MR in die äußere 5-polige Buchse gesteckt.
- 'MIDI-IN' des Rechners in die innere (mehr in Platinenmitte).

In den dargestellten Skizzen bezeichnet der dicke Punkt die Stelle, an der die Abschirmung des Kabels angelötet wird.

Besorgen Sie sich zur Herstellung der Kabel 2-polige oder 3-polige, abgeschirmte Leiter.

4.6.1 Herstellung von MIDI-Kabeln

Als MIDI-Kabel eignen sich in der Regel normale, abgeschirmte **Stereo**-Überspielkabel mit einer Länge von bis zu 2 Metern. **Mono**-Kabel sind nicht geeignet! Die Anschaffung von speziellen MIDI-Kabeln, wie sie oft in Musikgeschäften angeboten werden, lohnt sich nicht. Sie benötigen 2 MIDI-Kabel. Sie können sich die Kabel aber auch selbst leicht herstellen. Die Nummern der Steckerpins sind normalerweise ganz klein auf dem Kunststoffkörper aufgedruckt:



MIDI-Kabel, Aufsicht auf die Stecker von vorne

4.6.2 Herstellung von RS232-Kabeln

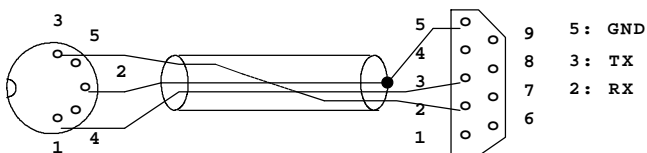
Je nach gewünschten Buchsentyten kommen für die Kommunikation über RS232 zwei verschieden Kabel zum Einsatz. Auch hier gilt: Die Nummern der Pins sind meistens ganz klein auf den Steckern aufgedruckt: Bei manchen Rechnern (vor allen Dingen älteren PCs) ist eine (oder auch beide) der seriellen Schnittstellen als

25-poliger Sub-D-Stecker ausgeführt. In diesem Fall entnehmen sie die Steckerbelegung ihrem PC-Handbuch oder besorgen sich im Fachhandel leicht erhältliche Übergangsadapter von 25-polig Sub-D auf 9-polig Sub-D. Die Kabellänge bei RS232 und einer Übertragungsrate von 9600 Baud (Default-Wert für MR) kann bis zu 10 Metern betragen. 'TX' bezeichnet die Sende-Daten-Leitung, 'RX' die Empfangs-Daten.

a.) RS232-Kabel: 9-polig Sub-D auf 5-polig DIN-Stecker

Dieses Kabel müssen Sie sich selbst herstellen, da es eine individuelle Beschaltung besitzt. Die Abschirmung kann bei Bedarf auf der Sub-D-Seite mit dem Steckergehäuse verbunden sein (ist aber nicht unbedingt nötig).

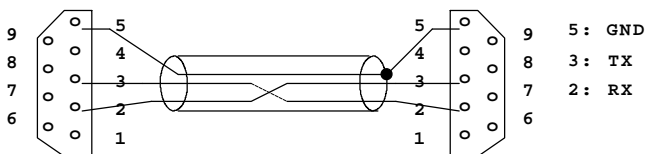
Dieses Kabel wird an MR in die innere DIN-Buchse auf MR eingesteckt.



RS232-Kabel: 9-polig Sub-D auf 5-polig DIN. Ansicht auf Stecker/bzw. Steck-Buchse von vorne.

b.) RS232-Kabel: 9-polig Sub-D auf 9-polig Sub-D

Bei diesem Kabel sind die TX- und RX-Adern gekreuzt. Dieses Kabel wird oft auch als "Nullmodem-Kabel" bezeichnet. Es ist zu verwenden, wenn sie anstelle der mitgelieferten 5-poligen DIN-Buchsen einen gewinkelten Sub-D-Stecker auf Ihrem MR eingelötet haben:



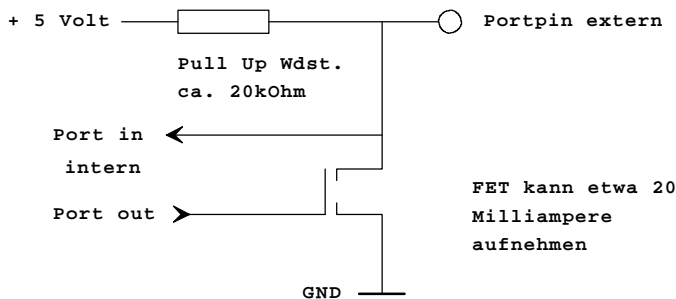
RS232-Kabel: 9-polig Sub-D auf 9-polig Sub-D. Ansicht auf Steck-Buchsen von vorne.

5 Zusatzhardware

Um ihnen die Entwicklung eigener Schaltungen etwas zu erleichtern möchten wir ihnen in diesem Kapitel einige Anregungen geben. Sie finden die Layouts, bzw. Schaltpläne der besprochenen Erweiterungen im Anhang.

5.1 Portstruktur der 51-er

Die Prozessoren der 51-er Familie besitzen sehr flexibel einsetzbare Ports. Für eigene Hardware-Entwicklungen ist die Kenntnis der Portstruktur hilfreich:



Vereinfachte Portstruktur der 51-er

- Nach Reset befinden sich Portpins immer auf High.
- **Eingabegeräte:** Taster, Schalter, ... werden immer zwischen dem Portpin des 51-ers und GND angeschlossen. Aus Schutzgründen können Sie dem Taster, Schalter, ..., einen 150 Ohm Widerstand vorschalten (Struktur des Ports ist in Wirklichkeit komplizierter, auch zwischen +5V und dem Pin liegt noch ein zweiter FET, der aber nur für wenige Mikrosekunden bei Schreibzugriffen auf den Portpin durchschaltet).
- **Ausgabe:** Im Low-Zustand ist der FET durchgeschaltet.

5.2 Basisplatine

Dieses Layout kann ihnen als Anregung und Ausgangsbasis bei eigenen Hardware-Entwürfen dienen. Auf dieser einseitigen Platine befindet sich ein 80C31 (oder auch 80C32) Minimalsystem mit 32kB-EPROM. Die Signale der I/O-Ports P1 und P3, sowie das Reset-Signal liegen an der Buchsenleiste an, über welche die Schaltung auch ihre Versorgungsspannung von 5 Volt erhält. Die Schaltung arbeitet auch zuverlässig mit den 16MHz-Versionen der 80C31/32.

6 Dies & das ...

In diesem Kapitel finden Sie diverse nützliche Informationen rund um dieses System.

6.1 Eigene Interrupts

Der 8031 springt bei Auftreten eines Interrupts immer auf eine fest vorgegebene Adresse (Für den 80535 kommen am Ende dieser Liste seine weiteren Interrupts dazu, siehe Literatur zum 80535. Das im folgenden für den 8031 beschriebene gilt genauso für den 80535):

```
$0003  Externer Interrupt 0 (Pin -INT0)
$000B  Timer 0 Interrupt
$0013  Externer Interrupt 1 (Pin -INT1)
$001B  Timer 1 Interrupt
$0023  Serieller Interrupt (Senden und Empfangen)
```

Im Entwicklungssystem MR liegen diese Adressen im EPROM, außerdem stehen für die einzelnen Interruptroutinen so nur je 8 Bytes zur Verfügung. Daher befindet sich an dieser Stelle im Entwicklungssystem-EPROM folgender Code:

```
$0000:  reset:          ljmp ...      ; Systeminitialisierung
$0003:  int0:           ljmp $8003
                          .ds.b 5
$000B:  timer0:        ljmp $800b
                          .ds.b 5
$0013:  int1:           ljmp $8013
                          .ds.b 5
$001B:  timer1:        ljmp $801b
                          .ds.b 5
$0023:  serial:       jb PSW.5,?ose ; F1=1: Seriell abfangen
                          ljmp $8023      ; F1=0: USER-Seriell
?ose:   ajmp ...
(Hier folgen die zusätzlichen Interrupts des 80C535)
```

Bei Auftreten eines Interrupts springt der 8031 also zur entsprechenden Adresse im RAM! Verwendet werden Interrupts z. Bsp. im Schrittmotorprogramm oder im Sprachsynthesizer.

Eine Besonderheit stellt der serielle Interrupt dar: Normalerweise dient er ja zur Steuerung des Entwicklungssystems. Wollen Sie jedoch den seriellen Interrupt für eigene Programme verwenden, müssen sie das **Prozessorflag F1** löschen, was natürlich auch heißt, daß dieses Flag anderweitig nicht verwendet werden darf!

Verwendet Ihr Programm Interrupts, und läuft es im RAM korrekt, und beginnt es normalerweise an Adresse \$8000, können Sie es ohne Probleme in ein EPROM oder in einen Single-Chip-Mikrocontroller brennen.

Allerdings laufen die Interrupts im EPROM etwas schneller ab, da ja nun jeweils ein Sprungbefehl fehlt!

6.2 Tips zur Fehlersuche

Über dieses Thema ließe sich natürlich sehr viel schreiben. Oft aber liegen Fehler an so elementaren Dingen, daß sie fast übersehen werden:

- **Initialisierung von Interrupts:** Interrupts werden durch 3 Bits initialisiert:
 - 1.) Das Enable-Bit des Interrupts im IE-Spezialfunktionsregister, (beim 80C535 IEN0, siehe nachfolgende Abschnitte)
 - 2.) Das Prioritäts-Bit des Interrupts im IP-Register,
 - 3.) Das generelle Interrupt-Enable- Bit EA, es ist das oberste Bit im Register IE (also IE.7). Speziell bei Timern gibt es auch noch das Zusammenspiel zwischen Timer-Overflow-Flag (TF0/1) und Timer-Run-Flag (TR0/1): immer wenn ein Timer die Grenze 255 -> 0 überschreitet wird sein overflow-Flag (TF0/1) gesetzt. TF0/1 wird bei der Ausführung eines Interrupts automatisch zurückgesetzt. Bevor Sie Timer-Interrupts verwenden, sollten Sie zusätzlich diese Sequenz in Ihre Software einbauen:
 - 1.) Timer Interrupt disablen,
 - 2.) TR-Bit löschen,
 - 3.) TF-Bit zurücksetzen (= auf 0).
- **Stack:** der Stack wächst nach oben! Vergessen Sie niemals ihn zu initialisieren. Das Entwicklungssystem und der Source-Level-Debugger benötigen etwa 16 Bytes auf dem Stack. Nach einem Reset zeigt der Stack auf die RAM-Adresse \$2F, also auf das oberste Byte des Bit-adressierbaren Feldes.
- **Read-Modify-Write:** Jedem Port ist noch ein Zwischenspeicher vorgeschaltet, der den Ausgabepegel des Ports speichert. Manche Befehle lesen nun den Wert des Port-Latches, andere wiederum den wirklichen Pegel am Port. Insbesondere im Zusammenhang mit Port 3 kann dies zu Problemen führen: Der Befehl 'mov A,P3' etwa liest den am Port **außen** anliegenden Pegel, indem einfach der Port auf LOW geschaltet wird, und dann gemessen wird, ob ein Strom fließt. Leider hängen auch die Signale -RD und -WR an diesem Port. Durch den Lesebefehl ist deshalb das RAM der Meinung, es soll beschrieben werden, was sofort zum Absturz von MR führt! Dagegen ist das Lesen des Portlatches völlig ungefährlich. Zu diesem Thema finden Sie Informationen unter dem Stichwort: Read-Modify-Write-Befehle in jedem Buch zum Thema 51-er Controller.

6.3 Eine 80C535 Inkompatibilität

Der in der MR-Hardware eingesetzte Controller 80C535 ist von seiner Leistungsfähigkeit her im oberen Teil der 51-er Familie anzusiedeln. Er deckt so ziemlich

alles ab, was seine kleineren Geschwister können. Daher werden Sie wohl des öfteren Programme mit MR entwickeln, die später auf kleineren Controllern (z. B. dem 8031) laufen sollen. Sie können dann normalerweise als Register-Definitions-Dateien direkt die Datei 'REG51.DEF' verwenden, mit einer Ausnahme:

Beim 8031 und einigen Verwandten sind Interrupt-Kontrollregister bei anderen Adressen als beim 80535!

Da der 80535 mehr Interrupt-Quellen als der 8031 besitzt, wurde das Interrupt-Kontroll-Register des 8031 'IE' auf die 80535-Register 'IEN0' und 'IEN1' verteilt, und, da sich die Priorität der einzelnen Interrupts in jeweils 4 Stufen einstellen lassen (im Gegensatz zum 8031 wo es nur 2 Prioritätsebenen gibt), wurde auch das 'IP'-Register des 8031 beim 80535 auf die beiden Register 'IP0' und 'IP1' verteilt. Leider sitzt beim 8031 das 'IP'-Register **genau** an der Adresse, an der im 80535 das 'IEN1'-Register sitzt.

Es gilt daher: Wenn Sie unter Verwendung der Register-Definitionsdatei 'REG51.DEF' auf einem 80C535 ein Programm für einen 8031 entwickeln, dürfen Sie keinesfalls das 'IP'-Register beschreiben. Denn anstelle daß sich die Priorität des gewünschten Interrupts ändert, würde ja in Wirklichkeit (weil eben in Wirklichkeit ja auch ein 80C535 in der Schaltung steckt) das 'IEN1'-Register des 80C535 beschrieben. Dies hätte zur Folge, daß der 80C535 selbständig unerwünschte Interrupts auslösen würde! Ansonsten verhält sich der 80C535 bei Einsatz der Register-Definitionsdatei 'REG51.INC' wirklich wie ein 8031!

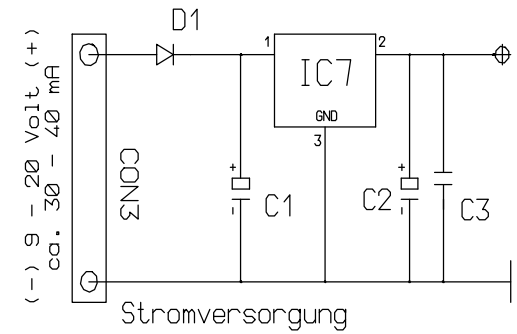
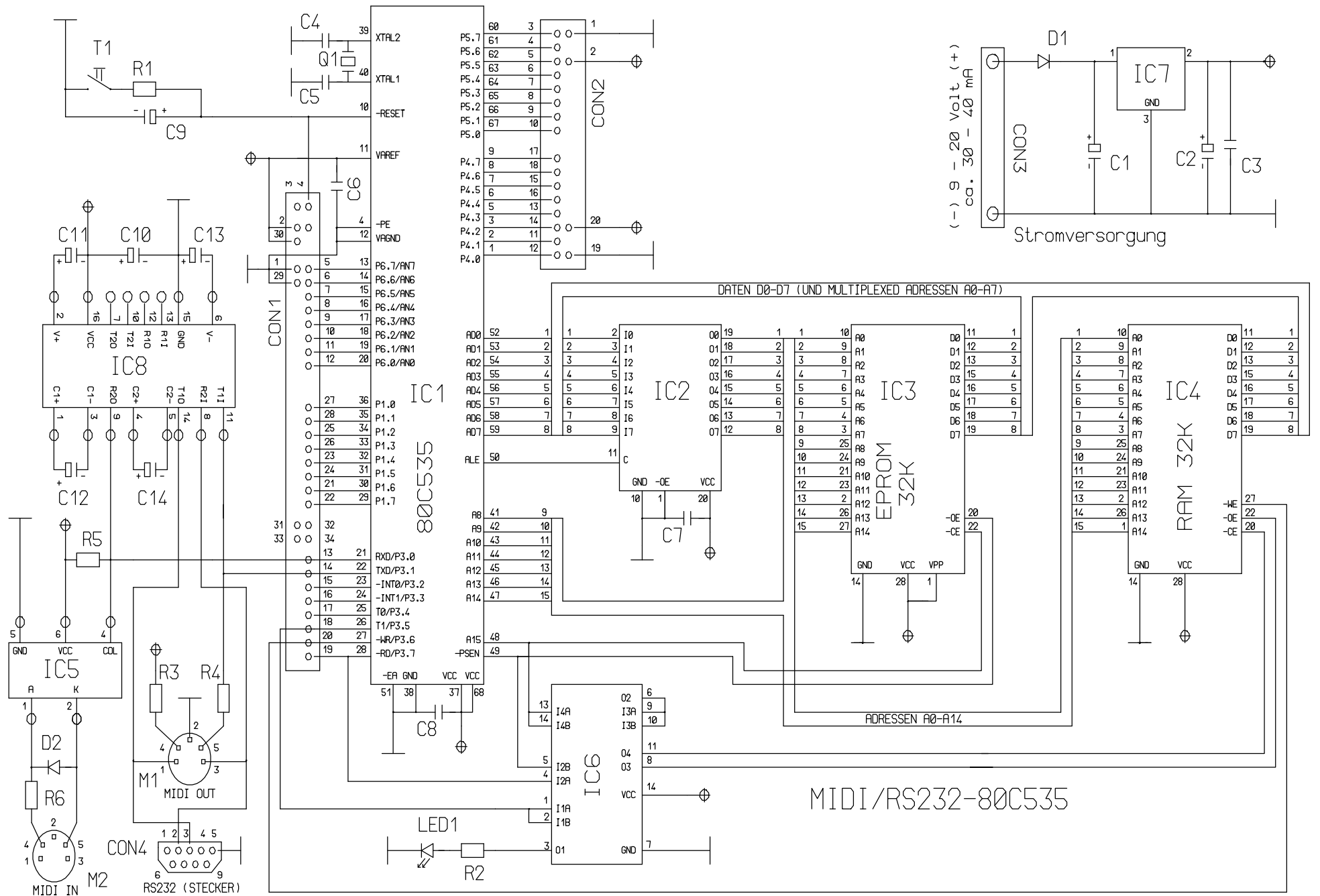
6.4 Watchdog des 80535

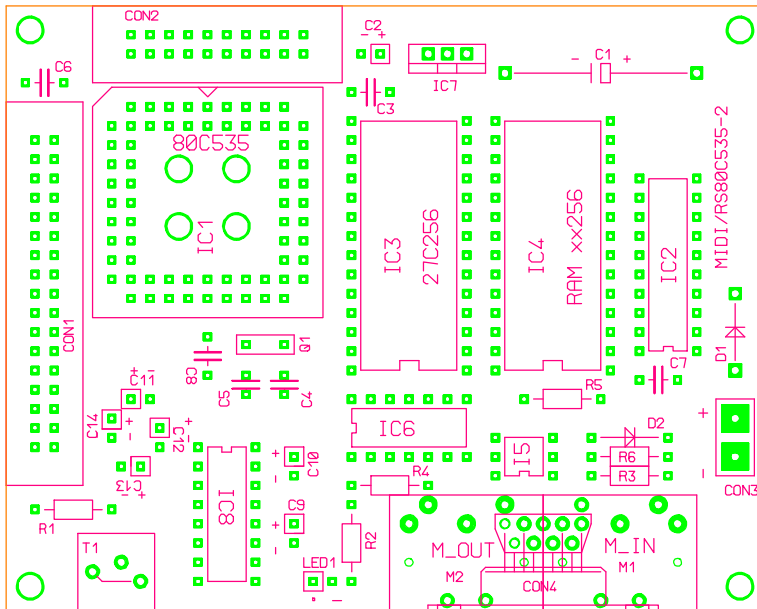
Der 80535 enthält eine Überwachungslogik ("Watchdog") zur Funktionskontrolle von Soft- und Hardware. Diese arbeitet nach folgendem System:

- Wird die Watchdog irgendwann innerhalb eines Programmes zum ersten Mal initialisiert, erwartet der Controller, daß das Programm in regelmäßigen Abständen zwei Bits in speziellen SF-Registern löscht.
- Unterbleibt die Löschung innerhalb dieses Zeitraumes (beträgt bei einem 80535 mit 12MHz etwa 65 Milisekunden) führt der Prozessor selbständig einen Software-Reset durch!
- Erfolgt nach diesem Software-Reset immer noch keine Programmreaktion, folgt der nächste Software-Reset, usw.
- Ein solcher "Dauer-Watchdog-Software-Reset" kann nur durch einen Hardware-Reset (Druck der Reset-Taste auf der MR-Platine oder Abschalten) beendet werden!

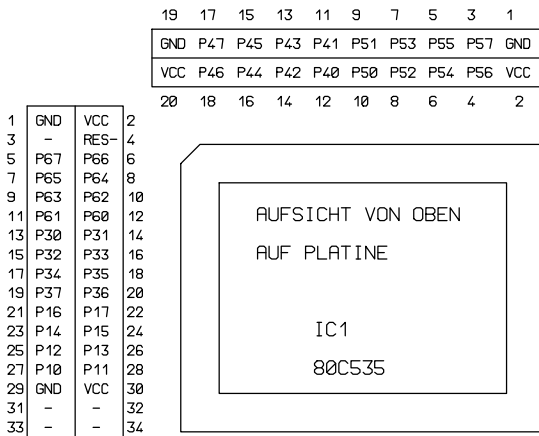
7 Anhang

- MIDI/RS232-80C535
 - Schaltplan
 - Bestückungsplan
 - Buchsenbelegung
 - IC-Tabelle
- Basisplatine
 - Schaltplan
 - Layout
 - Bestückungsplan

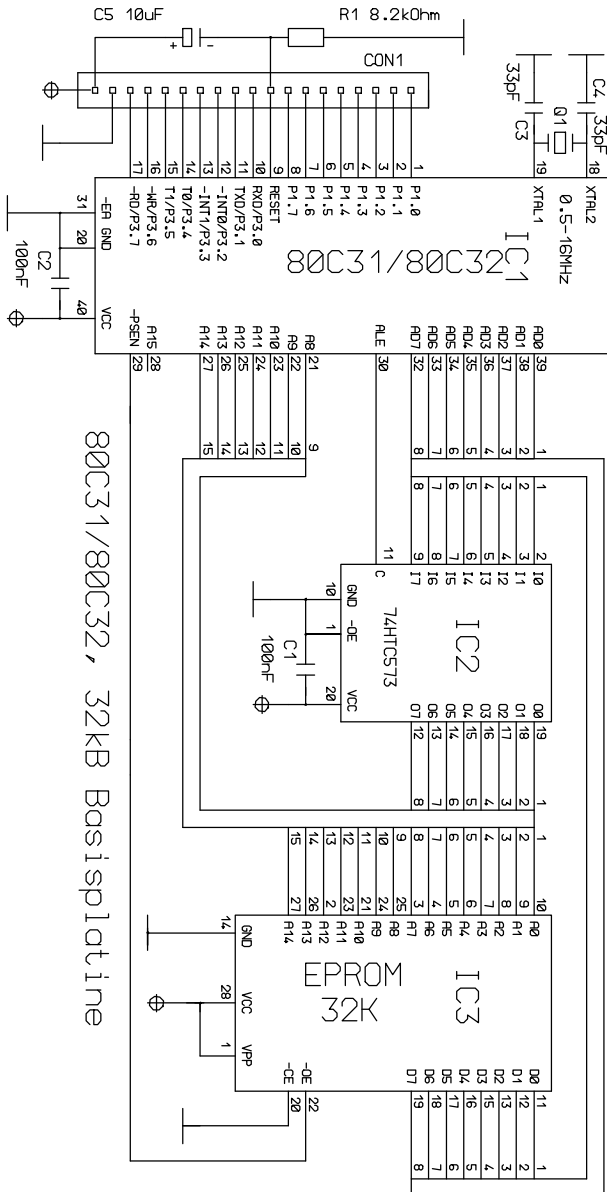




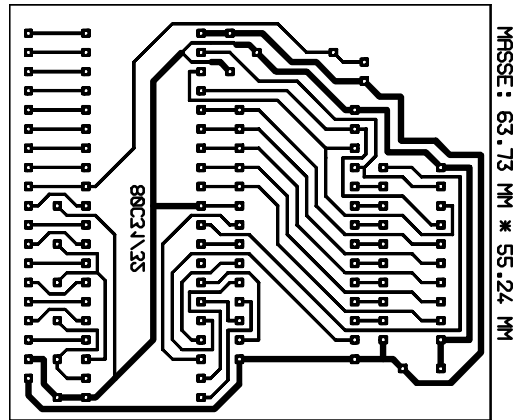
MIDI/RS232-80C535 Bestückungsplan



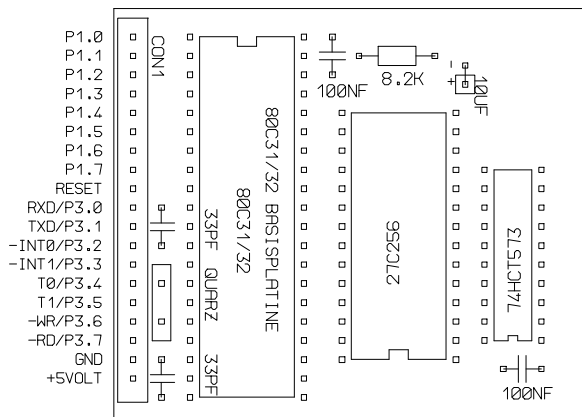
MIDI/RS232-80C535 Belegung Buchsenleisten



Basisplatine



Basisplatine Layout



Basisplatine Bestückungsplan

