



## USB-I/O-Interface UIO 88

**Viele Hobby-Programmierer haben gute Ideen für die Steuerung unterschiedlichster Vorgänge und Geräte mit dem PC. Meist jedoch fehlt es ihnen an (zeitgemäßer) Hardware, mit der z. B. Schaltvorgänge ausgeführt oder Zustände abgefragt werden können. Das USB-I/O-Interface stellt eine solche universelle Schnittstelle zur Verfügung. Das Board verfügt über 8 Eingänge und 8 Ausgänge, die sich einfach über die USB-Schnittstelle des PCs ansteuern bzw. abfragen lassen.**

### Fühler nach draußen

Viele Hobby-Programmierer nutzen ihren PC nicht nur als Schreib- oder Spielgerät, sondern realisieren die unterschiedlichsten Anwendungen selbst. Da werden z. B. Internetseiten und Datenbanken für den alltäglichen Gebrauch erstellt. Für diese reinen Software-Projekte wird in der Regel keine spezielle Hardware benötigt, so dass der erfahrene Programmierer sofort nach dem Starten des PCs mit seinen eigenen Projekten beginnen kann.

Geht es jedoch darum, in die Schnittstellen-Programmierung einzusteigen oder mit dem PC externe Vorgänge zu steuern und zu überwachen, dann kommt man um den Zukauf einer zusätzlichen, externen Hardware nicht mehr herum. In den vergange-

nen Jahren war die Programmierung der parallelen oder seriellen Schnittstelle der Regelfall, wobei diese sehr einfach beherrschbar sind. Zunehmend verdrängt aber der USB die althergebrachten Schnittstellen, beispielsweise verfügen viele tragbare Rechner gar nicht mehr über eine serielle Schnittstelle. Daher bietet sich die zeitgemäße USB-Schnittstelle für externe Steuerungen an.

Das hier vorgestellte USB-I/O-Interface UIO 88 verfügt über acht digitale Ein- und Ausgänge, mit denen bereits viele Standardaufgaben erledigt werden können. Beispielsweise lassen sich Lampen steuern, Zustände erfassen und vieles mehr realisieren. Die Ausgänge sind so dimensioniert, dass diese in beiden Zuständen (high oder low) einen Strom von 20 mA treiben können. Somit lassen sich beispiels-

### Technische Daten:

#### Eingänge:

Spannung: 0 V (low) bzw. 5 V (high)  
Max. Länge der Anschlussleitung: 3 m

#### Ausgänge:

Spannung: 0 V (low) bzw. 5 V (high)  
Max. Ausgangsstrom je Kanal  
(high und low): ..... 20 mA  
Max. Länge der Anschlussleitung: 3 m

**USB-Schnittstellen-Konfiguration:**  
9600 Baud, 8 Datenbits, gerade Parität,  
1 Stoppbit

#### Allgemein:

Spannungsversorgung: ..... über USB  
Max. Stromaufnahme: ..... 400 mA  
Gehäuseabm. (B x H x T): 58 x 24 x 120 mm  
Systemvoraussetzung: WIN 98, ME,  
XP, 2000, USB-Schnittstelle

weise externe Schaltgeräte wahlweise nach „high“ oder nach „low“ ansteuern.

## Beschaltung und Anschluss

Die Beschaltung der digitalen Ein- und Ausgänge muss der Anwender je nach geplanter Verwendung des USB-I/O-Interface UIO 88 individuell realisieren, wobei jedoch die Angaben in den technischen Daten einzuhalten sind. Ein entsprechendes Anwendungsbeispiel ist in Abbildung 1 zu sehen. Da kommen eingangsseitig Taster ebenso in Betracht wie Schaltausgänge von Geräten.

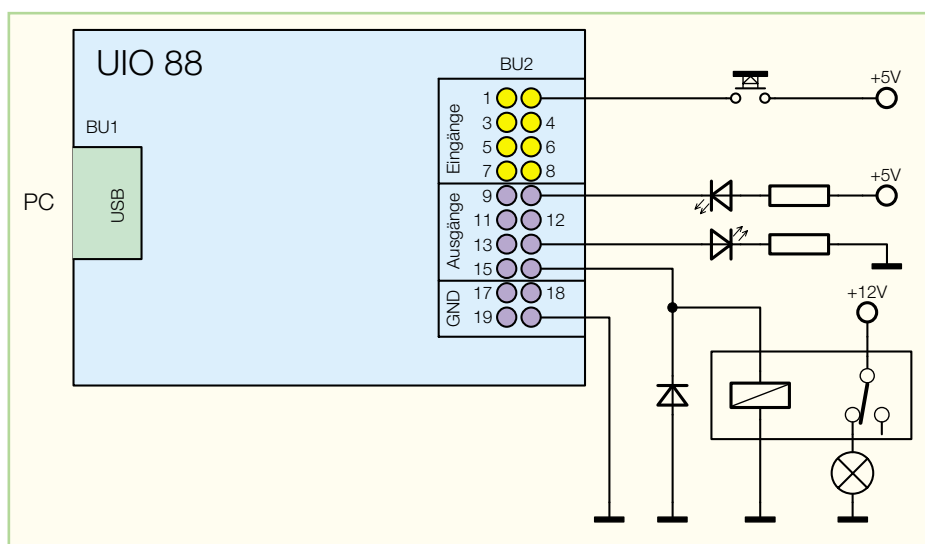
Auch ausgangsseitig sind vielfältige Steuerungsmöglichkeiten denkbar, von der einfachen LED über Schaltstufen, Relais bis hin zur Ansteuerung von Optokopplern und Geräte-Schalteneingängen. Die Spannungsversorgung der Baugruppe erfolgt über den USB, dadurch entfällt auch jedes Problem in Bezug auf Netzspannung. Die Verbindung zum PC wird mit einem normalen USB-Verbindungskabel hergestellt.

## Datenübertragung

Die USB-Datenübertragung funktioniert in dieser Schaltung ähnlich wie eine serielle Datenübertragung. Sie erfolgt mit einer Baud-Rate von 9600 bit/s, 8 Datenbit, gerader Parität und einem Stoppbit. Was genau hinter diesen Abkürzungen steckt, soll im Folgenden erläutert werden.

Die Geschwindigkeit der Datenübertragung wird üblicherweise in „bit/s“ angegeben. Diese Angabe kennzeichnet die Anzahl von einzelnen Bits, die pro Sekunde übertragen werden. Jedes übertragene Byte ist zusätzlich in einen entsprechenden Datenrahmen gefasst, der später noch genauer betrachtet wird, und der in diesem Fall aus drei zusätzlichen Bit besteht. Es werden also nicht nur die reinen „Nutzdaten“ übertragen. Aus diesem Grunde ist die effektiv übertragene Anzahl an Daten natürlich etwas geringer, als es die reine Angabe der Datenübertragungsgeschwindigkeit aussagt. Eine Datenübertragungsgeschwindigkeit von 9600 bit/s klingt für USB sehr langsam, ist aber für den Betrieb des UIO 88 mehr als ausreichend.

Tabelle 1: Verwendete ASCII-Steuerzeichen	
Zeichen	Wert
STX	02hex
ETX	03hex
ENQ	05hex
ACK	06hex
DC2	12hex
DC3	13hex
NAK	15hex



**Bild 1: Anwendungsbeispiele für die Ein- und Ausgangsbeschaltung des UIO 88**

Bei jeder Datenübertragung können Fehler auftreten, die jedoch durch verschiedene Fehlererkennungsmethoden zu identifizieren sind. Eine einfache und effektive Methode ist das Einfügen eines Paritätsbits, bei dem zwischen gerader (even parity) und ungerader Parität (odd parity) unterschieden wird. Der Sender untersucht hier das zu übertragende Byte und zählt dabei die Anzahl der logischen Einsen. Bei gerader Parität wird das Paritätsbit so (zurück-) gesetzt, dass die Gesamtzahl aller Einsen in Datenbyte und Paritätsbit gerade ist. Bei ungerader Parität wird das entsprechende Bit so (zurück-)gesetzt, dass die Gesamtzahl ungerade ist. Der Empfänger zählt ebenfalls die Anzahl aller Einsen und ermittelt daraus, ob ein Fehler bei der Datenübertragung aufgetreten ist. Diese Methode erkennt Fehler, bei denen in der Datenübertragung eine ungerade Anzahl von Bit (1, 3, 5, 7) falsch übertragen wurde. Diese Methode ist für einfache Datenübertragungen, wie sie beim UIO 88 verwendet wird, ausreichend. Sollen jedoch größere Datenmengen übertragen werden, so empfiehlt sich die softwaremäßige Implementation anderer effektiver Fehlererkennungsmethoden wie etwa die Prüfsummenbildung (z. B. CRC-16).

Wie bereits erwähnt, gibt es bei der seriellen Übertragung einen Datenrahmen, der durch ein Start- und ein Stoppbit die Kombination aus Daten und Paritätsbit umrahmt. Er dient dazu, dass der Empfänger den Beginn eines Datenblockes eindeutig erkennt und dann die übertragenen Bits entsprechend erfasst. Jeder Datenblock beginnt mit einer logischen „1“ als Startbit und endet mit einer logischen „0“ als Stoppbit. Die Anzahl der Stoppbit ist beim UIO 88 auf eines festgelegt, allerdings können andere Geräte auch mit 1,5 oder 2 Stoppbit arbeiten.

Bis jetzt haben wir erst die unterste Ebene der Datenübertragung erläutert, das verwendete Protokoll des UIO 88 besteht jedoch pro Befehl aus mehreren zu übertragenden Bytes und benutzt außerdem einen Protokollrahmen. Jedes Datenpaket beginnt mit dem Steuerzeichen „STX“ (02hex) und endet mit dem Zeichen „ETX“ (03hex). An diesen Zeichen kann der jeweilige Empfänger genau erkennen, wann ein Datenpaket beginnt bzw. endet, und weiß somit, wann ein Befehl oder dessen Antwort komplett empfangen wurde. Kommen diese Steuerzeichen jedoch in den Daten vor, so müssen sie entsprechend umgeformt werden, um einen vorzeitigen Neustart bzw. Abbruch der Kommunikation zu verhindern. Die Konvertierung der Zeichen wird entsprechend folgender Auflistung durchgeführt (eine Übersicht der verwendeten ASCII-Zeichen befindet sich in Tabelle 1):

<STX>	→	<ENQ>	<DC2>
<ETX>	→	<ENQ>	<DC3>
<ENQ>	→	<ENQ>	<NAK>

Taucht eines der drei Zeichen „STX“, „ETX“ oder „ENQ“ in den Daten auf, ist es also durch die entsprechende Zeichenfolge auszutauschen. Empfängerseitig muss diese Umsetzung jedoch wieder rückgängig gemacht werden, um die eigentlichen Nutzdaten wiederherzustellen. Der entsprechende Befehlssatz des UIO 88 ist in Tabelle 2 zu sehen und dort erläutert.

Als Beispiel für die beschriebene Vorgehensweise sollen der Ausgang 2 auf high, alle anderen gleichzeitig jedoch auf low gesetzt werden. Den entsprechenden Befehl „o“ entnimmt man Tabelle 2 und ergänzt mit den entsprechenden Parametern sowie dem Protokollrahmen:

<STX>	o	<02hex>	<ETX>
-------	---	---------	-------

Tabelle 2: Befehlssatz des UIO 88

<b>Abfragen der Eingänge:</b>	
Befehl:	„I“
Parameter:	keine
Antwort:	aktueller Zustand der Eingänge (1 Byte), ACK (1 Byte)
<b>Abfragen aller Ausgänge:</b>	
Befehl:	„O“
Parameter:	keine
Antwort:	aktueller Zustand der Ausgänge (1 Byte), ACK (1 Byte)
<b>Setzen aller Ausgänge:</b>	
Befehl:	„o“
Parameter:	auszugebende Daten (1 Byte)
Antwort:	Ausgangszustand nach dem Setzen (1 Byte), ACK (1 Byte)
<b>Setzen eines Ausgangs:</b>	
Befehl:	„s“
Parameter 1:	Nummer des Ausgangs (0–7, 1 Byte),
Parameter 2:	gewünschter Zustand (0 bzw. 1, 1 Byte)
Antwort:	Ausgangszustand aller Ausgänge (1 Byte), ACK (1 Byte)

Hier ist jedoch bereits zu sehen, dass der Befehlsparameter das Steuerzeichen „STX“ (02hex) enthält. Es muss also eine Konvertierung erfolgen:

```
<STX> o <ENQ> <DC2> <ETX>
```

Dieser Datenstring ist jetzt über die USB-Schnittstelle zu übertragen. Sind diese Bildungsregeln erst einmal verinnerlicht, stellt die Programmierung der einzelnen Befehle kein Problem mehr dar.

Zum besseren Verständnis der Ansteuerung des USB-I/O-Interface befindet sich auf der mit dem Bausatz mitgelieferten Programmdiskette ein Visual-C++-Beispielprojekt mit zugehörigen Quelltexten. Außerdem enthält die Diskette noch Informationen zur Programmierung einer USB-Schnittstelle.

Weitere Infos und Beispielprojekte zum Umgang mit der USB-Schnittstelle lassen sich auf der Internet-Seite des USB-Chip-Herstellers „www.ftdichip.com“ abrufen.

### Schaltung

Die Schaltung des USB-I/O-Interface (UIO 88) ist in Abbildung 2 und 3 dargestellt. Die Schaltung der Ein- und Ausgänge (Abbildung 2) erscheint auf den ersten Blick recht umfangreich, ist aber dennoch schnell zu überblicken, da sich die Ein- und Ausgangsstufen jeweils 8fach wiederholen.

Das zentrale Element des UIO 88 bildet der Mikrocontroller IC 2. Dieser erfasst die Daten von den digitalen Eingängen ST 1 bis ST 8, wandelt diese Daten um, wertet die Befehle der USB-Schnittstelle aus und steuert die digitalen Ausgänge ST 9 bis ST 16 an.

Der interne Oszillator wird durch den Quarz Q 2 und die beiden Kondensator C 22 und C 23 auf eine Frequenz von 14,745 MHz stabilisiert. Der Schaltungsteil mit C 24, R 14 und D 3 sorgt für einen definierten Reset-Impuls beim Zuschalten der Betriebsspannung und damit für definierte Verhältnisse des Mikrocontrollers beim Einschalten oder nach einem Spannungsausfall. Eine weitere externe Beschaltung ist zum normalen Betrieb des Mikrocontrollers nicht notwendig. Der Mikrocontroller erhält seine Befehle über das vorgeschaltete USB-Modul („USB-Umsetzung“). Dieses USB-Modul setzt ankommende Befehle vom USB-Format in serielle Befehle und auch abgehende serielle Befehle in das USB-Format um, so dass der Mikrocontroller diese an seinen Schnittstellenpins (RXD und TXD) „verstehen“ kann.

Das zentrale Element des USB-Moduls (Abbildung 3) bildet der USB-Controller (IC1/FTU232AM), der speziell für die Konvertierung zwischen USB und RS 232 entwickelt wurde. Dieses IC beinhaltet auch eine Art Mikrocontroller, so dass eine Taktversorgung gewährleistet sein muss. Der interne Oszillator wird mit dem Quarz Q 1 und den Kondensatoren C 12 und C 13 auf eine Frequenz von 6 MHz stabilisiert. Ein definiertes Starten des USB-Controllers wird durch Beschalten des Reset-Pins mit T 1, R 6, R 7, R 8 und C 8 sichergestellt.

Zur Speicherung der USB-Erkennungsdaten (Vendor-ID, Product-ID, Seriennummer etc.) des UIO 88 ist an die „Microwire“-Schnittstelle von IC 1 ein EEPROM vom Typ 93C46 (IC 3) angeschlossen.

Neben den beiden Leitungen für die Betriebsspannung besteht der USB aus zwei Datenleitungen (D+, D-). Diese sind jeweils über einen Widerstand zum Leitungs-

abschluss (R 1, R 3) mit dem USB-Controller IC 1 verbunden.

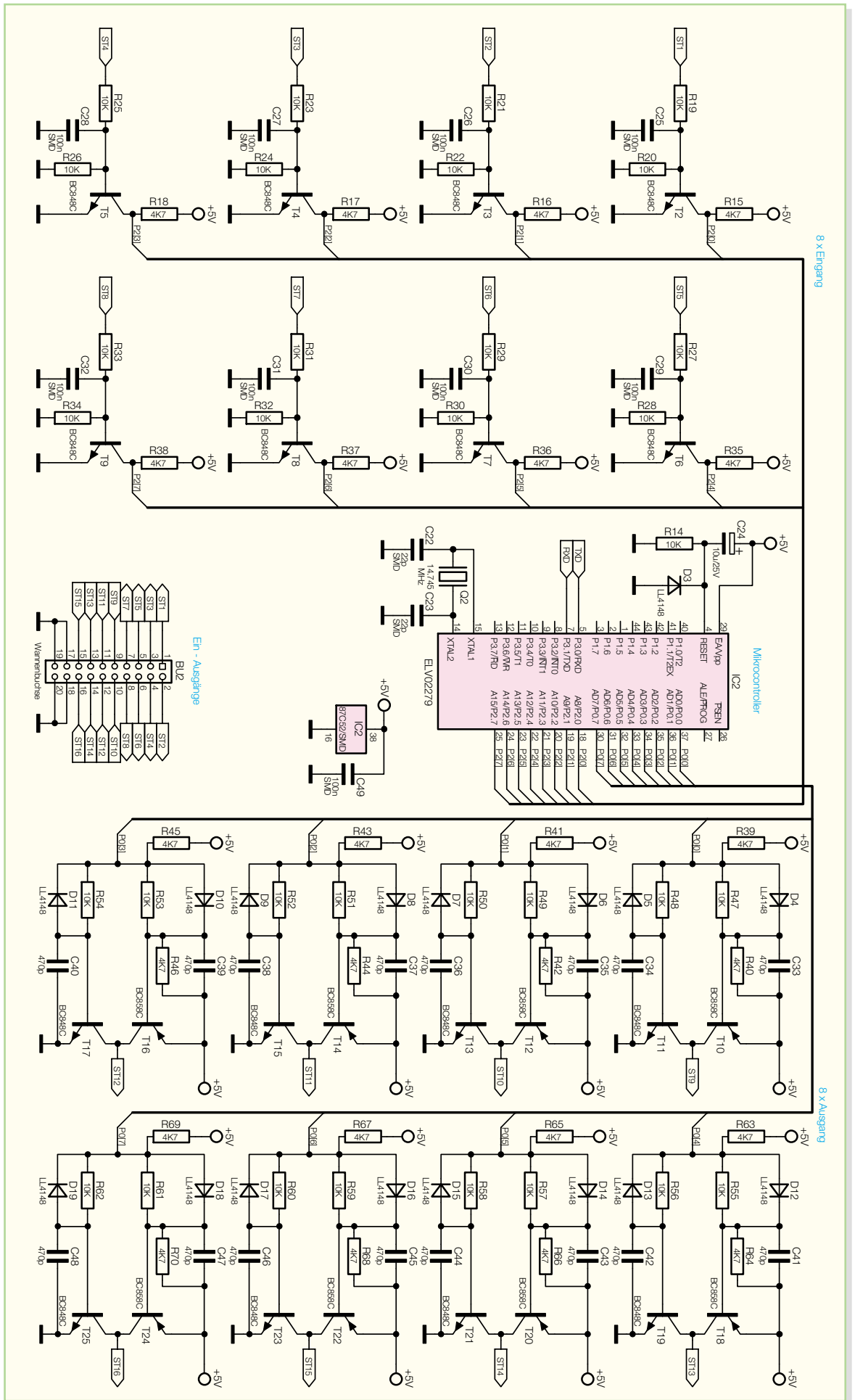
Der Widerstand R 2 dient als Pull-up-Widerstand des USB. Über den definierten „High“-Pegel der D+-Datenleitung erkennt der PC eine angeschlossene UIO-88-Hardware. Die Leuchtdioden D 1 und D 2 dienen als Statusanzeige für die Übertragung, wobei die grüne LED (D 1) einen Sendevorgang und die rote LED (D 2) einen Datenempfang der UIO-88-Hardware signalisiert.

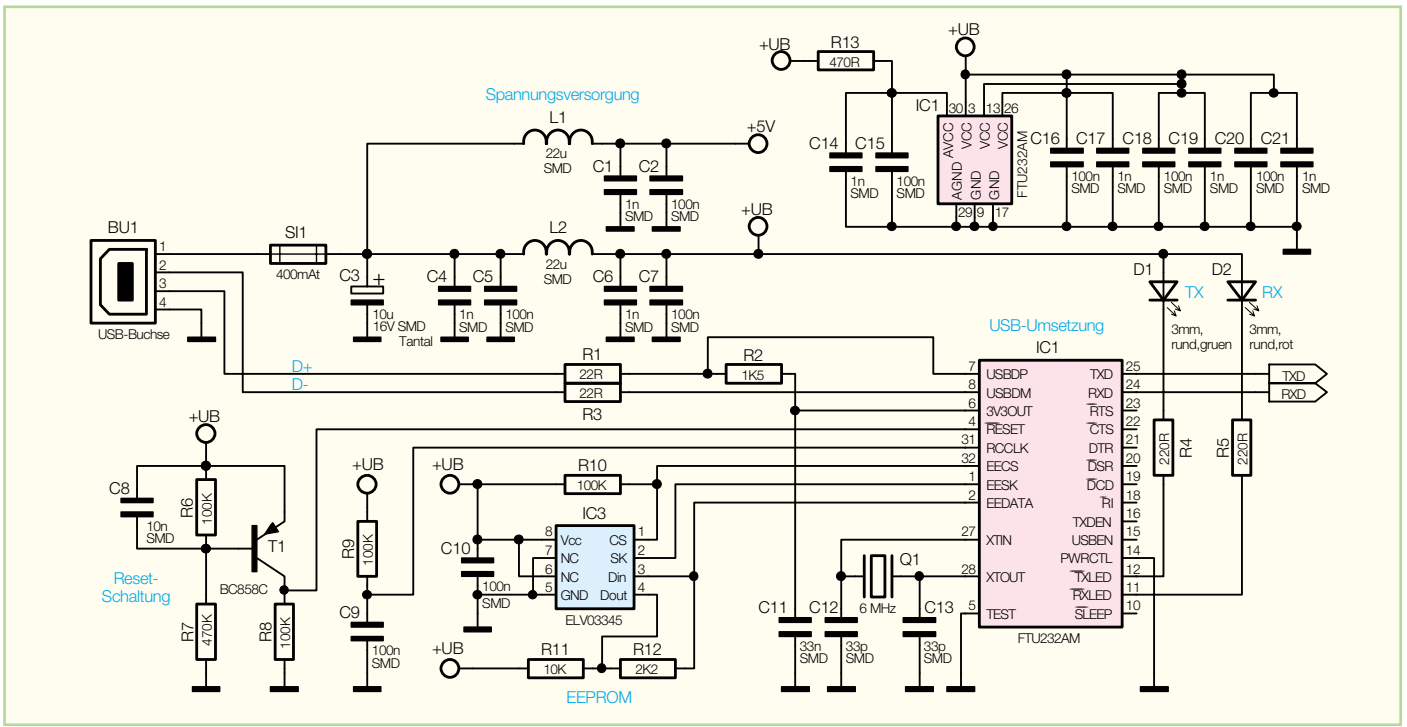
Die Spannungsversorgung der gesamten Schaltung erfolgt über den USB, der eine Spannung von 5 V zur Verfügung stellt. Diese Betriebsspannung gelangt über die USB-Buchse (Pin 1 und Pin 4) auf die Schaltung. Die Kondensatoren C 1 bis C 7 sowie die Induktivitäten L 1 und L 2 dienen zur Stabilisierung dieser Spannung bzw. zur Unterdrückung hochfrequenter Störungen. Die Sicherung SI 1 ist als Überlastungsschutz des USB implementiert.

Das UIO 88 verfügt über 8 Eingänge und 8 Ausgänge, die sich jeweils schaltungstechnisch nicht unterscheiden. Aus diesem Grund wird jeweils nur ein Ein- bzw. Ausgang beschrieben. Der an Eingang ST 1 angelegte Pegel steuert über einen Widerstandsteiler, bestehend aus R 19 und R 20, den Transistor T 2 an. Dieser invertiert das Signal und gibt es weiter an den Mikrocontroller. Der Kondensator C 25 parallel zum Widerstand R 20 dient zur Unterdrückung eventuell auftretender Störungen.

Die Ausgänge sind ein wenig aufwändiger, da hier entsprechende Ströme getrieben werden sollen, und das sowohl bei High- als auch bei Low-Pegel. Die Endstufe besteht deshalb also aus zwei Transistoren, von denen immer nur einer durchgeschaltet sein darf, um einen Kurzschluss der Betriebsspannung mit der Schaltungsmasse zu verhindern. Hierfür kommt am Ausgang ST 9 ein PNP-Transistor (T 10) und ein NPN-Transistor (T 11) zum Einsatz. Diese werden vom Mikrocontroller über eine Portleitung, aber über unterschiedliche Vorwiderstände angesteuert. Der Transistor T 10 wird leitend, sobald die Basisspannung ca. 0,7 V geringer ist als die Betriebsspannung, der Transistor T 11 wird leitend, sobald die Basisspannung ca. 0,7 V beträgt. Somit ergibt sich ein Spannungsbereich, in dem beim Umschaltvorgang beide Transistoren kurzzeitig im leitenden Zustand sind und somit einen zeitlich begrenzten Kurzschluss verursachen. Aus diesem Grunde erfolgt das jeweilige Zuschalten der Transistoren leicht verzögert. Die Verzögerung wird durch die beiden Kondensatoren C 33 und C 34 realisiert. Die Dioden D 4 und D 5 schließen den entsprechend parallel liegenden Widerstand kurz, um den Verzögerungskondensator beim Abschalten des zugehö-

**Bild 2: Die Ein- und Ausgänge des USB-I/O-Interface UIO 88**





**Bild 3: Die USB-Umsetzung des USB-I/O-Interface UIO 88**

rigen Transistors schneller zu entladen. Durch diese Maßnahme wird erreicht, dass die einzelnen Transistoren verzögert einschalten, aber schnell wieder abschalten, so dass sich zu keinem Zeitpunkt beide Transistoren im leitenden Zustand befinden können.

### Nachbau

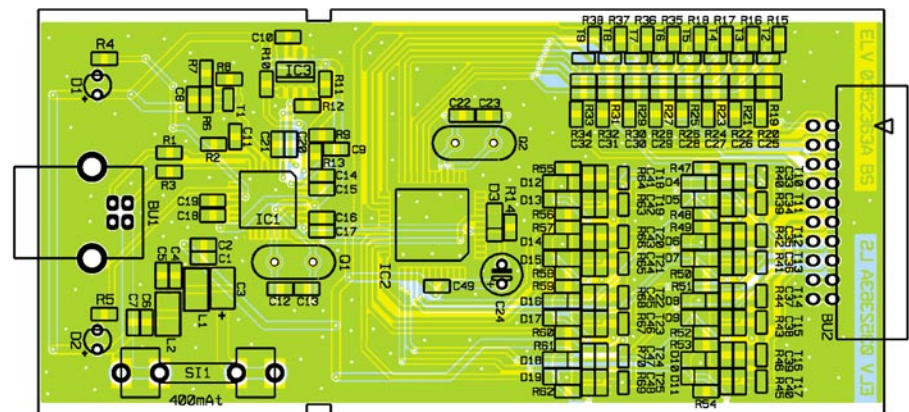
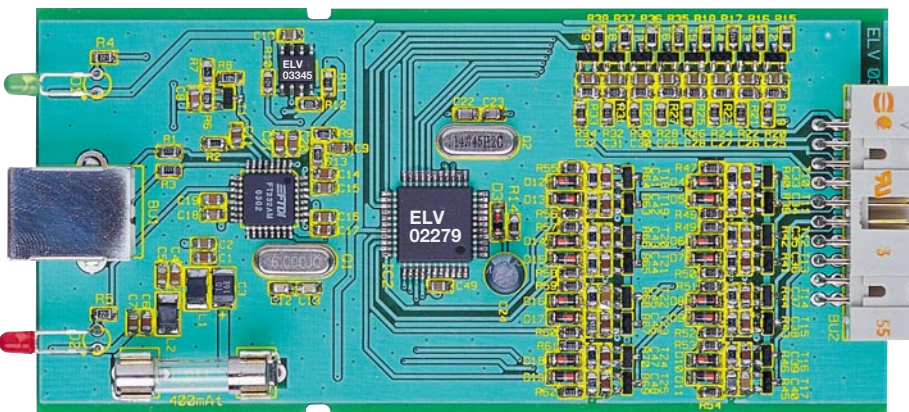
Der Nachbau des USB-IO-Interface

UIO 88 erfordert ein wenig Geschick, da die verwendeten Bauelemente fast ausschließlich in SMD-Technik ausgeführt sind, um ein kompaktes Design zu erreichen. Neben einem geregelten Lötcolben mit sehr feiner Spitze, SMD-Lötzinn sowie Entlötlitze sollte auch eine SMD-Pinzette zum Positionieren der kleinen Bauteile nicht fehlen. Auch eine starke und möglichst beleuchtbare Standlupe leistet hier gute Dienste. Der Aufbau erfolgt an-

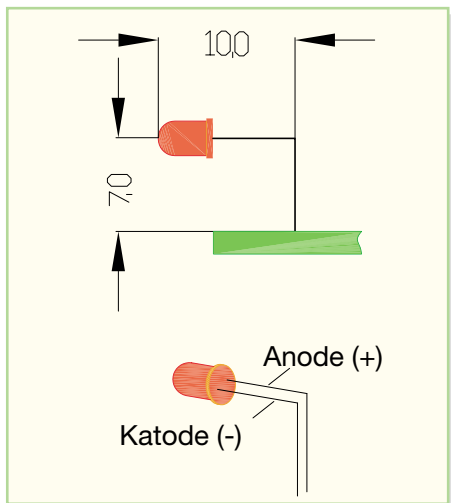
hand des Bestückungsdrucks, des Platinenfotos sowie der Stückliste.

Er beginnt mit den ICs 1, 2 und 3. Diese haben einen sehr geringen Pin-Abstand und sind am besten zu bestücken, wenn ringsum noch keine Bauteile die Lötarbeiten behindern. Beim Bestücken dieser Bauteile ist besonders auf die korrekte Einbaulage zu achten, da nachträgliche Korrekturen nur sehr schwer durchführbar sind und dabei (nicht nur im Hobbylabor) meist Platine und/oder Bauelement beschädigt werden können. Beim USB-Controller (IC 1) und dem Mikrocontroller (IC 2) ist die Pin 1 zugeordnete Ecke entweder angeschragt oder durch eine kreisförmige Ausfräsung des Gehäuses gekennzeichnet, die sich auch im Bestückungsdruck wiederfindet. Bei IC 3 ist die Pin 1 zugeordnete Seite abgeflacht bzw. durch eine Gehäusekerbe gekennzeichnet. Bei den ICs wird zunächst jeweils ein Lötpad vorverzinnt, an dem diese zuerst verlötet werden. Im Anschluss daran ist ein zweiter Pin an der diagonal gegenüberliegenden Seite zu verlöten. Dabei ist darauf zu achten, dass alle Anschlüsse des ICs auf den zugehörigen Löt-pads aufliegen, um später Kontaktfehler durch ungenügende Verlötung auszuschließen. Bevor die weiteren Anschlüsse mit der Leiterplatte verlötet werden, ist nochmals die richtige Position zu überprüfen.

Nach dem Verlöten aller IC-Pins und sorgfältiger Kontrolle auf Kurzschlüsse (überflüssiges Zinn mit feiner Entlötlitze absaugen) geht es nun an die weiteren SMD-Komponenten, die SMD-Widerstände, -Kondensatoren und -Spulen. Hier wird wieder zunächst jeweils ein Löt-pad auf der Leiterplatte vorverzinnt, bevor man das Bauteil mit der Pinzette erfasst, positioniert und am vorverzinnten Pad anlötet. Nach der Kontrolle der korrekten



**Ansicht der fertig bestückten Platine des UIO 88 mit zugehörigem Bestückungsplan**



**Bild 4: So erfolgt die Präparation und Bestückung der LEDs.**

Position des Bauteils ist der zweite Anschluss zu verlöten. Die Kondensatoren sollten erst direkt vor dem Bestücken einzeln aus der Verpackung genommen werden, da diese keinen Aufdruck tragen, der über den Wert informiert. Als Nächstes erfolgt die Bestückung des SMD-Elkos C 3. Bei SMD-Elkos ist üblicherweise der +-Pol gekennzeichnet. Im Anschluss daran sind die SMD-Transistoren und -Dioden in gleicher Weise auf der Leiterplatte zu verlöten. Hier ist jedoch besonders auf die richtige Polung zu achten, die sich bei den Transistoren aus der Pin-Konfiguration ergibt. Die Dioden sind durch eine Ring-Markierung an der Katode gekennzeichnet.

Vor der weiteren Bestückung sind alle SMD-Lötstellen sorgfältig zu kontrollieren, ggf. unter Zuhilfenahme einer starken Lupe. Ist alles in Ordnung, beginnt die Bestückung der konventionell bedrahteten

Bauelemente. Bei diesen Bauteilen ist darauf zu achten, dass überstehende Drahtenden auf der Lötseite der Platine mit einem Elektronik-Seitenschneider so abgetrennt werden, dass einerseits die Lötstelle nicht beschädigt wird, andererseits hervorstehende Drahtenden keine Kurzschlüsse im späteren Betrieb hervorrufen können. Die beiden Quarze (Q 1 und Q 2) und der Elko sind in stehender Position zu bestücken und zu verlöten. Bei dem Elko C 24 ist unbedingt auf richtige Polung zu achten, da dieser sonst im schlimmsten Fall sogar explodieren könnte. Üblicherweise sind diese am Minuspol gekennzeichnet.

Die beiden Buchsen BU 1 und BU 2 und der Sicherungshalter müssen direkt auf der Leiterplatte aufliegen, bevor die Anschlüsse verlötet werden, da die Lötstellen bei späterer mechanischer Belastung der Buchsen ziemlich stark beansprucht werden.

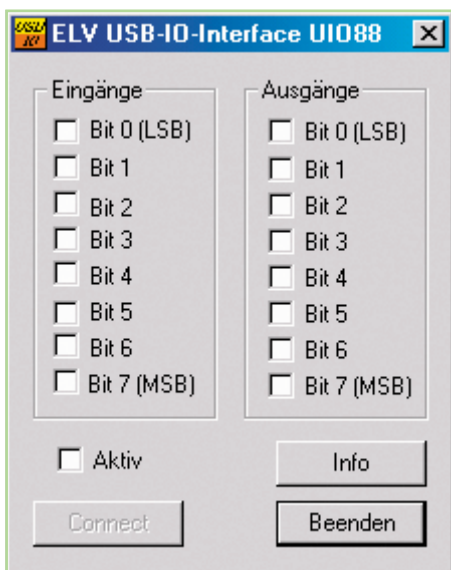
Nun erfolgt die polrichtige Bestückung der Leuchtdioden, hier ist die Anode durch den längeren Anschlusspin gekennzeichnet. Die LEDs sind nach etwa 10 mm um 90 Grad abzuwinkeln und dann etwa 7 mm hoch zu verlöten, so dass keine Probleme bei der Montage des Gehäuses auftreten (siehe Abbildung 4).

Abschließend, nach nochmaliger Überprüfung der Leiterplatte auf Bestückungsfehler und Lötzinnbrücken, ist das Gehäuse zu montieren. Hierfür wird die Platine mit der Seite der USB-Buchse und den LEDs zuerst in die Gehäuseoberseite eingelegt, so dass die Platine in die „Gehäuse-Nasen“ einrastet. Daraufhin ist die Gehäuseunterschale von der Seite her aufzuschieben.

### Funktionskontrolle

Nach Abschluss des Aufbaus ist das USB-I/O-Interface mittels des mitgelieferten Testprogramms einfach zu überprüfen. Dazu verbindet man das I/O-Interface über ein USB-Verbindungskabel mit einem PC. Dieser sollte die neu angeschlossene Hardware erkennen und nach kurzer Zeit einen USB-Treiber verlangen. Dieser Treiber (ftd2xx.inf) befindet sich auf der mitgelieferten Programmdiskette.

Anschließend ist das Testprogramm „uio88\_test.exe“ von der mitgelieferten Programmdiskette zu starten. In dem daraufhin erscheinenden Dialogfeld (Abbildung 5) ist der Punkt „Aktiv“ einzuschalten. Jetzt fragt das Programm die Eingänge des I/O-Interface in einem festen Zeitraster ab. Der linke Fensterbereich stellt den Zustand der Eingänge entsprechend dar. Im selben Zeitraster steuert das Programm die Ausgänge. Durch Setzen der entsprechenden Häkchen im rechten Fensterbereich werden die Ausgänge entweder gesetzt oder zurückgesetzt, was mit einem Multimeter



**Bild 5: Das Testprogramm für das USB-I/O-Interface**

### Stückliste: USB-I/O-Interface UIO 88

#### Widerstände:

22 Ω/SMD .....	R1, R3
220 Ω/SMD .....	R4, R5
470 Ω/SMD .....	R13
1,5 kΩ/SMD .....	R2
2,2 kΩ/SMD .....	R12
4,7 kΩ/SMD .....	R15–R18,
	R35–R46, R63–R70
10 kΩ/SMD .....	R11, R14, R19–R34,
	R47–R62
100 kΩ/SMD .....	R6, R8–R10
470 kΩ/SMD .....	R7

#### Kondensatoren:

22 pF/SMD .....	C22, C23
33 pF/SMD .....	C12, C13
470 pF/SMD .....	C33–C48
1 nF/SMD .....	C1, C4, C6, C14,
	C17, C19, C21
10 nF/SMD .....	C8
33 nF/SMD .....	C11
100 nF/SMD .....	C2, C5, C7, C9,
	C10, C15, C16, C18, C20,
	C25–C32, C49
10 µF/16V/SMD .....	C3
10 µF/25V .....	C24

#### Halbleiter:

FT8U232AM/SMD .....	IC1
ELV02279/SMD .....	IC2
ELV03345/SMD .....	IC3
BC858C .....	T1, T10, T12, T14,
	T16, T18, T20, T22, T24
BC848C .....	T2–T9, T11, T13,
	T15, T17, T19, T21, T23, T25
LL4148 .....	D3–D19
LED, 3mm, grün .....	D1
LED, 3mm, rot .....	D2

#### Sonstiges:

Quarz, 6 MHz, HC49U4 .....	Q1
Quarz, 14,745 MHz, HC49U4 .....	Q2
SMD-Induktivität, 22 µH .....	L1, L2
USB-B-Buchse, winkelpoint .....	BU1
Wannen-Steckleiste, winkelpoint,	
2 x 10-polig .....	BU2
Sicherung, 0,4 A, träge .....	SI1
Platinensicherungshalter (2 Hälften),	
print .....	SI1
1 3,5"-Diskette UIO 88	
1 Profil-Gehäuse, blau, bearbeitet	
und bedruckt	

kontrolliert werden sollte. Jetzt kann der Test der Eingänge erfolgen. Dazu legt man nacheinander High-Pegel an jeden Eingang. Sofern der Eingang in Ordnung ist, ist das jeweils zugehörige Feld im Anzeigefenster der PC-Software markiert.

Funktionieren alle Ein- und Ausgänge, ist das Modul bereit für eigene Applikationen. 