



USB-AD-Wandler UAD 8

Die Messwertaufnahme und -verarbeitung stellt einen wichtigen Bereich in der Elektronik dar, in dem der PC immer öfter zum Einsatz kommt. Der USB-AD-Wandler (UAD 8) bietet sich als Verbindungsglied zur analogen Außenwelt an, mit der analoge Spannungen für den PC über eine USB-Schnittstelle digital umgesetzt werden.

Der UAD 8 stellt acht AD-Wandler-Eingänge zur Verfügung, womit sich analoge Spannungen im Bereich von 0 bis 2,5 V umsetzen lassen. Durch Vorschalten entsprechender Spannungsteiler lassen sich auch höhere Spannungen erfassen.

Messen via USB

Eigene Applikationen am PC zu entwickeln, ist dank komfortabler Programmiersprachen auch für den Hobby-Programmierer relativ einfach. Sollen diese mit der Außenwelt kommunizieren, erfolgt dies über die Standard-Schnittstellen des PCs, früher nahezu ausschließlich über die Parallel- oder serielle Schnittstelle, heute zunehmend via USB-Port. Besonders die Messwertaufnahme und -verarbeitung ist das Gebiet der Elektronik, in dem der PC in immer mehr Anwendungen eingesetzt wird. Beispielsweise sollen mit einem PC unterschiedliche Spannungen gemessen oder eine spezielle „Sensor-Spannung“ aus-

gewertet werden. Denn der PC macht das Sammeln, Verarbeiten und Visualisieren von Daten besonders einfach. Allerdings ist zur Aufnahme von analogen Daten eine externe Hardware notwendig, die die Wandlung der analogen Werte in digitale Daten vornimmt.

Der USB-AD-Wandler (UAD 8) bietet sich hierbei als Verbindungsglied zwischen Rechner und analoger Außenwelt an. Der Anschluss an den PC wird über eine USB-Schnittstelle vorgenommen. Der UAD 8 stellt acht AD-Wandler-Eingänge zur Verfügung, womit sich analoge Spannungen im Bereich von 0 bis 2,5 V umsetzen lassen. Durch Vorschalten eines Spannungsteilers lassen sich auch höhere Spannungen messen. Die Kanäle 1 bis 6 arbeiten mit

einer Auflösung von 10 Bit, die Kanäle 7 und 8 mit einer Auflösung von 8 Bit.

Der Einsatz des UAD 8 hat auch für die Programmierung der Applikation einen entscheidenden Vorteil: Der Programmierer muss sich nicht mit den Besonderheiten der USB-Schnittstellenprogrammierung, die es bekanntermaßen „in sich“ hat, beschäftigen, er braucht lediglich das später beschriebene Datenübertragungsprotokoll in seine Applikation einbinden.

Der USB-AD-Wandler

Beschaltung und Anschluss

Die Beschaltung der AD-Wandler-Eingänge ist vom Anwender je nach geplanter Verwendung des UAD 8 individuell zu

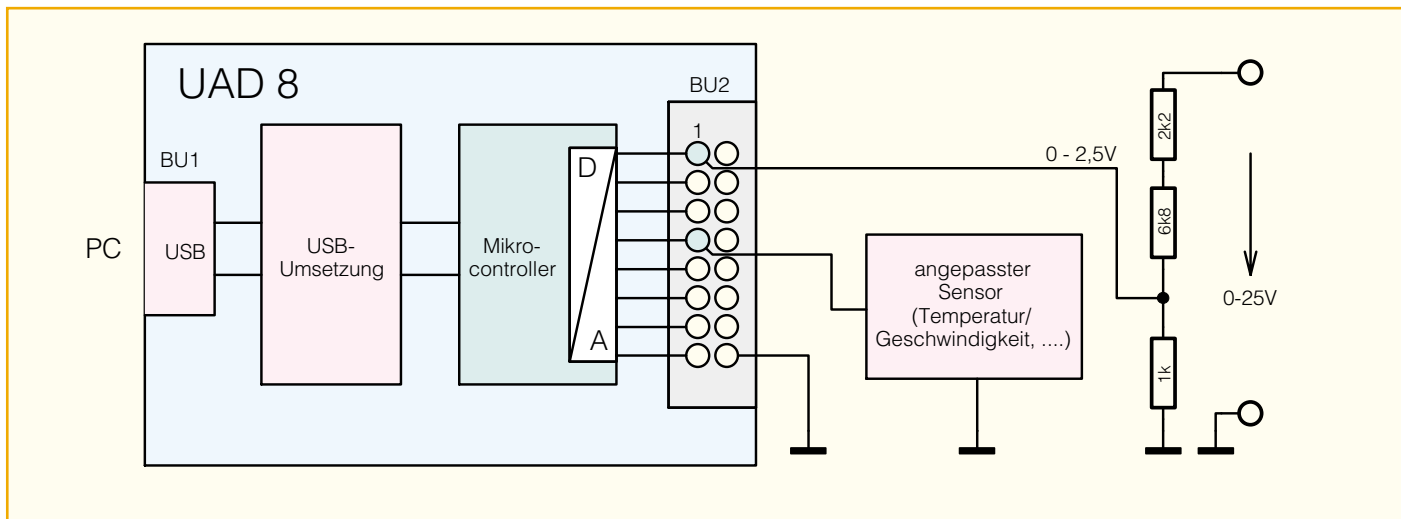


Bild 1: Beispiel für die Beschaltung des AD-Wandlers

gestalten, wobei natürlich die Angaben in den technischen Daten einzuhalten sind. Ein entsprechendes Anwendungsbeispiel ist in Abbildung 1 zu sehen. Da kommen eingangsseitige Spannungsteiler zur Spannungsüberwachung ebenso in Betracht wie Sensoren oder die Schaltausgänge von Geräten. Der hieraus erzeugte digitale Wert der Spannung wird durch das jeweils angepasste Protokoll in die eigene Applikation übertragen und dort ausgewertet, z. B. visualisiert. Die Spannungsversorgung der Schaltung erfolgt über den USB (Universal Serial Bus), dadurch entfällt auch jedes Problem in Bezug auf die Gefährdung und Einflüsse durch Netzspannung. Die Verbindung zum PC wird mit einem normalen USB-Verbindungskabel hergestellt.

Datenübertragung

Die USB-Datenübertragung funktioniert beim USB-AD-Wandler ähnlich einfach wie bei einer seriellen Datenübertragung, die vielen Programmierern von RS 232 her geläufig sein dürfte. Sie erfolgt mit einer Baudrate von 38.400 kBit/s, 8 Datenbits, gerader Parität und einem Stopbit. Die Bestandteile dieses Protokolls sollen im

Folgenden detailliert 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 (bestehend aus 8 Bit) ist zusätzlich in einen entsprechenden Datenrahmen gefasst, der später noch genauer betrachtet wird und der in diesem Fall aus drei zusätzlichen Bytes 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.

Bei jeder Datenübertragung können Fehler auftreten, die sich durch verschiedene Fehlererkennungsmethoden finden lassen. 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 Bits (1, 3, 5, 7) falsch übertragen wurde. Sie ist für eine einfache Datenübertragung, wie sie beim UAD 8 verwendet wird, ausreichend.

Wie bereits erwähnt, gibt es bei der seriellen Übertragung einen Datenrahmen, der durch ein Start- und ein Stopbit die Kombination aus Datenbyte und Paritätsbit im Wortsinn 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 Stopbit. Die Anzahl der Stopbits ist beim UAD 8 auf eines festgelegt, jedoch können andere Geräte auch mit 1,5 oder 2 Stopbits arbeiten.

Bis jetzt haben wir lediglich die unterste Ebene der Datenübertragung betrachtet, das verwendete Protokoll des UAD 8 besteht jedoch pro Befehl aus mehreren zu übertragenden Bytes und benutzt außerdem einen Protokollrahmen. Jedes Datenpaket beginnt mit dem Steuerzeichen (siehe Tabelle 1) „STX“ (02hex) und endet mit dem Zeichen „ETX“ (03hex). An die-

Technische Daten:

Anzahl der Analogeingänge:	8 unabhängige Kanäle
Auflösung:	10 Bit (Kanal 1 bis Kanal 6); 8 Bit (Kanal 7 und Kanal 8)
Messspannung:	0–2,5 V/DC
Genauigkeit:	± 1 %
Eingangswiderstand:	1 MΩ
Max. Länge der Anschlussleitung:	3 m
USB-Schnittstellen-Konfiguration:	38.400 Baud, 8 Datenbits, gerade Parität, 1 Stopbit
Spannungsversorgung:	über USB
Max. Stromaufnahme:	40 mA
Gehäuseabmessungen (B x H x T):	58 x 24 x 69 mm
Systemvoraussetzung:	(MS Windows 98/ME/XP/2000)

Zeichen	Wert
STX	02hex
ETX	03hex
ACK	06hex
NAK	15hex

Tabelle 1: Verwendete Steuerzeichen

Bedeutung	Befehl	Parameter	Antwort	Parameter 1	Parameter 2	Parameter 3	Parameter 4
Konfiguration	'k' (6Bhex)	Konfigurationswort (siehe Tabelle 3)	Bestätigung	ACK/NAK	---	---	---
Abfrage CH 1	'1' (31hex)	---	Messwert CH 1 (mV)	1000er	100er	10er	1er
Abfrage CH 2	'2' (32hex)	---	Messwert CH 2 (mV)	1000er	100er	10er	1er
Abfrage CH 3	'3' (33hex)	---	Messwert CH 3 (mV)	1000er	100er	10er	1er
Abfrage CH 4	'4' (34hex)	---	Messwert CH 4 (mV)	1000er	100er	10er	1er
Abfrage CH 5	'5' (35hex)	---	Messwert CH 5 (mV)	1000er	100er	10er	1er
Abfrage CH 6	'6' (36hex)	---	Messwert CH 6 (mV)	1000er	100er	10er	1er
Abfrage CH 7	'7' (37hex)	---	Messwert CH 7 (mV)	1000er	100er	10er	1er
Abfrage CH 8	'8' (38hex)	---	Messwert CH 8 (mV)	1000er	100er	10er	1er

Tabelle 2: Befehlssatz des UAD 8

sen Zeichen kann der jeweilige Empfänger genau erkennen, wenn ein Datenpaket beginnt bzw. endet, und „weiß“ somit, wann er einen Befehl oder dessen Antwort komplett empfangen hat. Die Parameter werden als ASCII-Zeichen übertragen, d. h. eine „0“ wird als Zeichen '0' (30hex), eine „1“ als Zeichen '1' (31hex), ... und eine „9“ als Zeichen '9' (39hex) übertragen. Dies hat den Vorteil, dass die Befehle und empfangenen Daten nicht mit Steuerzeichen verwechselt werden und somit einen vorzeitigen Neustart bzw. Abbruch der Kommunikation verursachen.

Übertragungsbeispiel

Zum besseren Verständnis des verwendeten Protokolls wird im Folgenden ein Übertragungsbeispiel dargestellt.

Der Befehlssatz des UAD 8 ist in Tabelle 2 dargestellt und erläutert.

Als Beispiel für den Umgang mit dem Befehlssatz soll die angelegte Spannung vom Kanal 3 und Kanal 5 digitalisiert und übertragen werden. Zuerst ist das Konfigurationswort zu senden. Die Tabelle 3 gibt an, dass die Aktivierung von Kanal 3 und Kanal 5 ein Konfigurationswort von

Kanal	Konfigurationswort
1	xxxx xxx1
2	xxxx xx1x
3	xxxx x1xx
4	xxxx 1xxx
5	xxx1 xxxx
6	xx1x xxxx
7	x1xx xxxx
8	1xxx xxxx

1 = aktiviert 0 = deaktiviert

Tabelle 3: Konfigurationswort

14hex (0001 0100b) ergibt. Dieser Parameter wird um 30hex (Zeichen '0') aufaddiert, um bei der Übertragung des Konfigurationswortes nicht in den Bereich der Steuerzeichen zu geraten, was zu einem vorzeitigen Abbruch oder Neubeginn der Übertragung führen würde.

Der Konfigurationsbefehl lautet dann:

```
<STX> k <44hex> <ETX>
```

Die erfolgreiche Konfiguration wird mit einem:

```
<STX> <ACK> <ETX>
```

bestätigt.

Das Abfragen der gemessenen Daten von Kanal 3 wird mit dem Befehl '3' (33hex) gestartet. Das Protokoll dazu setzt sich wie folgt zusammen:

```
<STX> <33hex> <ETX>
```

Als Antwort wird der Messwert aufgeteilt in 1000er, 100er, 10er und 1er mV übertragen. Eine Antwort eines Messwertes von 1,741 V sieht dann so aus:

```
<STX> <31hex> <37hex><34hex> <31hex> <ETX>
```

Für die weitere Bearbeitung ist von den empfangenen Parametern jeweils das Zeichen '0' (30hex) abzuziehen und der Dezimalpunkt, je nach Beschaltung des Eingangsspannungsteilers, zu setzen.

Die Abfrage des Kanals 5 gestaltet sich ähnlich. Der Befehl lautet:

```
<STX> <35hex> <ETX>
```

Ein Messwert von 0,642 V an Kanal 5 wird wie folgt übertragen:

```
<STX> <30hex> <36hex> <34hex> <32hex> <ETX>
```

Die Abfrage eines nicht konfigurierten Kanals wird mit:

```
<STX> <NAK> <ETX>
```

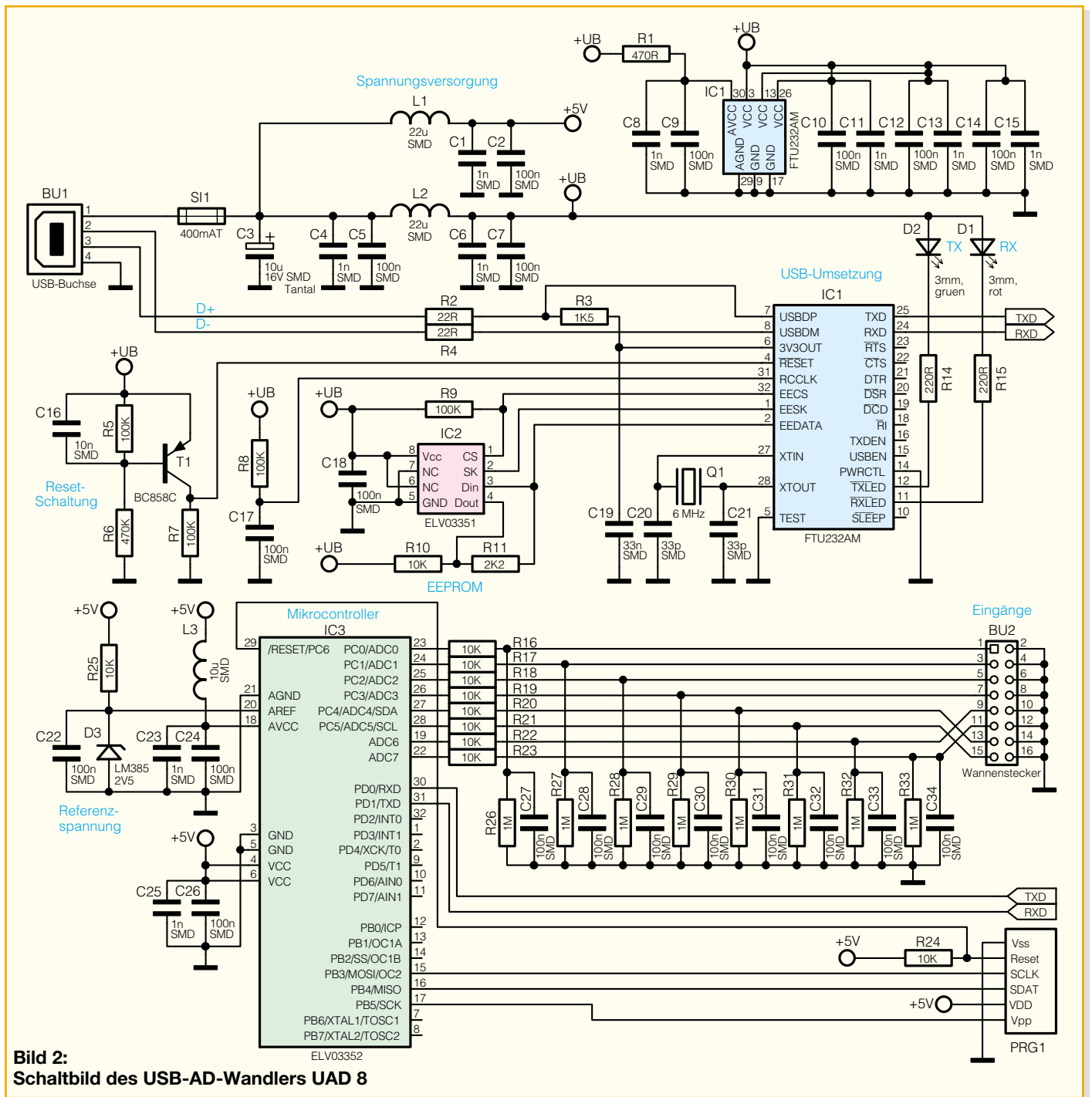
bestätigt.

Diese zusammengesetzten Datenstrings sind jeweils über die USB-Schnittstelle zu übertragen. Sind diese Bildungsregeln erst einmal verinnerlicht, sollte die Programmierung der einzelnen Befehle keine größeren Probleme mehr darstellen.

Zum besseren Verständnis der Ansteuerung des USB-AD-Wandlers befindet sich auf der dem Bausatz beiliegenden Programmdiskette ein Visual-C++-Beispielprojekt mit zugehörigen Quelltexten. Außerdem enthält die Diskette noch Informationen zur Programmierung einer USB-Schnittstelle. Zusätzlich lassen sich auf der Internetseite des USB-Chip-Herstellers „www.ftdichip.com“ noch weitere Infos zum Umgang mit der USB-Schnittstelle abrufen.

Schaltung

Die gesamte Schaltung des USB-AD-Wandlers (UAD 8) ist in Abbildung 2 dargestellt. Der Mikrocontroller IC 3 bildet das zentrale Element des UAD 8. Dieser wandelt die analogen Spannungen der AD-Wandler-Eingänge Kanal 0 (ADC 0) bis Kanal 7 (ADC 7) in digitale Daten um und wertet die Befehle der USB-Schnittstelle aus. Die Kondensatoren C 23 bis



C 26 und die Spule L 3 dienen zum Ausfiltern von Störungen der Betriebsspannung des internen AD-Wandlers von IC 3. Durch eine Verwendung des internen Oszillators kann auf eine diesbezügliche äußere Beschaltung verzichtet werden.

Der Widerstand R 25 und die Diode D 3 erzeugen eine stabile Referenzspannung von 2,5 V für den AD-Wandler. Der parallel geschaltete Kondensator C 22 dient zur Störunterdrückung dieser Referenzspannung.

Über einen Wannenstecker (BU 2) gelangen die analogen Mess-Spannungen an den Schaltungseingang. Die Widerstände R 16 bis R 23 realisieren eine Strombe-

grenzung und somit einen Schutz der AD-Wandler-Eingänge. Die Widerstände R 26 bis R 33 bilden den Eingangswiderstand für den jeweiligen AD-Wandler-Eingang.

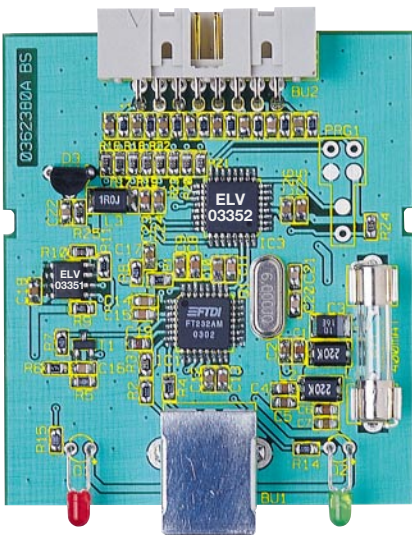
Der Programmieradapter PRG1 und der Widerstand R 24 ermöglichen eine Programmierung des Mikrocontrollers in der Serienproduktion.

Der Mikrocontroller empfängt und sendet 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 Schnittstellen-Pins

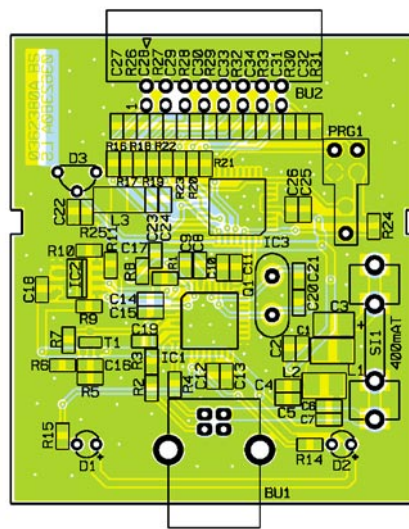
(RXD und TXD) „verstehen“ und „sprechen“ kann.

Kommen wir zur genaueren Beschreibung dieser USB-Umsetzung im oberen Teil des Schaltbildes. Das zentrale Element bildet hier der USB-Controller (IC 1, FTU232AM), der speziell für die Konvertierung zwischen USB und RS 232 entwickelt wurde. Dieser USB-Controller beinhaltet einen Mikrocontroller, der eine Taktversorgung bedingt. Der interne Oszillator wird mit dem Quarz Q 1 und den Kondensatoren C 20 und C 21 auf eine Frequenz von 6 MHz stabilisiert.

Durch Beschalten des Reset-Pins mit der Reset-Schaltung, bestehend aus T 1,



Ansicht der fertig bestückten Platine des USB-AD-Wandlers UAD 8 mit zugehörigem Bestückungsplan



R 5 bis R 7 und C 16, ist ein definiertes Starten des USB-Controllers sichergestellt.

Zur Speicherung der USB-Erkennungsdaten (Vendor-ID, Product-ID, Seriennummer etc.) des UAD 8 ist an die „Micro-wire“-Schnittstelle von IC 1 ein EEPROM vom Typ ELV 03351 (IC 2) angeschlossen. Diese Daten sichern u. a. die USB-typische Plug-&-Play-Erkennung durch das PC-Betriebssystem.

Neben den beiden Leitungen für die Betriebsspannung (dazu später mehr) besteht der USB aus zwei Datenleitungen (D+, D-). Diese sind jeweils über einen Widerstand zum Leitungsabschluss (R 2, R 4) mit dem USB-Controller IC 1 verbunden.

Der Widerstand R3 dient als Pull-up-Widerstand des USB. Über den definierten High-Pegel der D+-Datenleitung erkennt der PC die angeschlossene UAD-8-Hardware. Die Leuchtdioden D1 und D2 dienen als Statusanzeige für die Übertragung, wobei die grüne LED (D 2) einen Sendevorgang und die rote LED (D 1) einen Datenempfang 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 Sicherung SI 1 sichert den USB gegen eine Überlastung ab. 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 weiteren Kondensatoren (C8 bis C15) sind zur Stör- und Schwingneigungsunterdrückung des USB-Controllers (IC 1) implementiert.

Nachbau

Der Nachbau des UAD 8 erfordert ein wenig Geschick, da die verwendeten

Bauelemente fast ausschließlich in SMD-Technik ausgeführt sind, um ein möglichst kompaktes Design zu erreichen. Neben einem geregelten LötKolben 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 anhand des Bestückungsdrucks, des Platinenfotos sowie der Stückliste.

Er beginnt mit den ICs 1, 2 und 3. Diese haben einen geringen Pin-Abstand und sind am besten zu bestücken, wenn ringsum noch keine Bauteile die Lötarbeiten behindern. Beim Bestücken dieser Bauteile ist auf die korrekte Einbaulage zu achten. Beim USB-Controller (IC 1) und dem Mikrocontroller (IC 3) ist die Pin 1 zugeordnete Ecke entweder angeschrägt oder durch eine kreisförmige Ausfräsung des Gehäuses gekennzeichnet, welche sich auch im Bestückungsdruck wiederfindet. Bei IC 2 ist die Pin 1 zugeordnete Seite abgeflacht bzw. durch eine Gehäusekerbe gekennzeichnet.

Es wird zunächst jeweils ein Löt-pad vorverzinnt, an dem das lagerichtig aufgelegte IC zuerst mit einem Pin zu verlöten ist. Im Anschluss daran ist ein zweiter Pin an der diagonal gegenüberliegenden Seite zu verlöten. Dabei ist darauf zu achten, dass alle Anschlüsse des ICs plan auf den zugehörigen Löt-pads aufliegen, um spätere Kontaktfehler durch ungenügende Verlötung auszuschließen. Bevor die weiteren Anschlüsse mit der Leiterplatte verlötet werden, ist nochmals die richtige Position zu überprüfen, da eine Korrektur im Nachhinein nur noch sehr schwer durchführbar ist. Sollte durch zu viel aufgetragenes Löt-zinn einmal eine Lötbrücke zwischen den Pins entstanden sein, kann man diese sehr

**Stückliste:
USB-AD-Wandler UAD 8**

Widerstände:

22 Ω/SMD	R2, R4
220 Ω/SMD	R14, R15
470 Ω/SMD	R1
1,5 kΩ/SMD	R3
2,2 kΩ/SMD	R11
10 kΩ/SMD	R10, R16–R25
100 kΩ/SMD	R5, R7–R9
470 kΩ/SMD	R6
1 MΩ/SMD	R26–R33

Kondensatoren:

33 pF/SMD	C20, C21
1 nF/SMD	C1, C4, C6, C8, C11, C13, C15, C23, C25
10 nF/SMD	C16
33 nF/SMD	C19
100 nF/SMD	C2, C5, C7, C9, C10, C12, C14, C17, C18, C22, C24, C26–C34
10 µF/16 V/SMD	C3

Halbleiter:

FT8U232AM/SMD	IC1
ELV03351/SMD	IC2
ELV03352/SMD	IC3
BC858C	T1
LM385-2,5V	D3
LED, 3 mm, rot	D1
LED, 3 mm, grün	D2

Sonstiges:

Quarz, 6 MHz, HC49U4	Q1
SMD-Induktivität, 22 µH	L1, L2
SMD-Induktivität, 10 µH	L3
USB-B-Buchse, winkelprint	BU1
Wannen-Steckleiste, winkelprint, 2 x 8-polig	BU2
Sicherung, 0,4 A, träge	SI1
Platinensicherungshalter (2 Hälften), print	SI1
1 3,5"-Diskette UAD-8-Software	

einfach durch den Einsatz feiner Entlötlitze beseitigen.

Nach dem Verlöten der IC-Pins und folgender Kontrolle der Lötstellen geht es nun an die weiteren SMD-Komponenten –

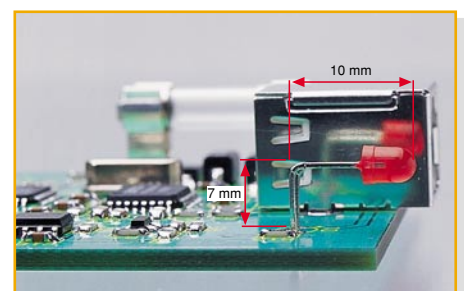


Bild 3: So erfolgt die LED-Bestückung.

die SMD-Widerstände, -Kondensatoren und -Spulen. Hier wird zunächst wiederum jeweils ein Lötpad auf der Leiterplatte vorverzinnt, bevor man das Bauteil mit der Pinzette erfasst, positioniert und am vorverzinnten Pad anlötet. Nach der Kontrolle der korrekten 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. Der SMD-Elko C 3 trägt üblicherweise an seinem Pluspol eine Markierung!

Im Anschluss daran ist der SMD-Transistor in gleicher Weise mit der Leiterplatte zu verlöten. Auch hier ist auf die richtige Polung zu achten, die sich allerdings bei den SMD-Transistoren aus der Pinkonfiguration ergibt.

Vor der weiteren Bestückung sind alle SMD-Lötstellen sorgfältig zu kontrollieren, ggf. unter Zuhilfenahme einer starken Lupe. Sind alle SMD-Komponenten aufgelötet und überprüft, beginnt die Bestückung der konventionell bedrahteten Bauelemente. Der Quarz (Q 1) und die Z-Diode (D3) sind in stehender Position zu bestücken und verlöten. Die beiden Buchsen BU 1 und BU 2 und der Sicherungshalter müssen direkt plan auf der Leiterplatte aufliegen, bevor die Anschlüsse verlötet werden, da die Lötstellen sonst bei späterer mechanischer Belastung der Buchsen ziemlich stark beansprucht werden. Durch die plane Auflage wird der

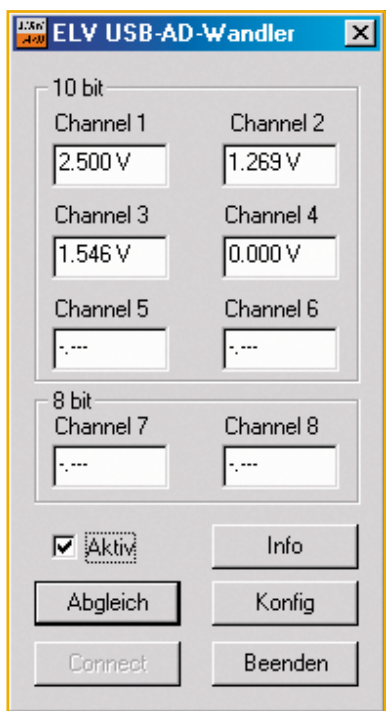


Bild 4: Das mitgelieferte Testprogramm hilft bei Abgleich und Konfiguration des AD-Wandlers.

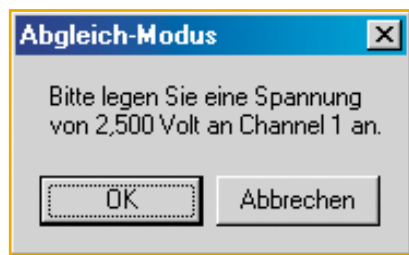


Bild 5: Der Abgleich-Modus des USB-AD-Wandlers

Großteil der wirkenden Kräfte auf die Platine selbst übertragen.

Nun erfolgt die polrichtige Bestückung der Leuchtdioden, hier ist die Anode durch den längeren Anschlusspin gekennzeichnet. Die LED-Anschlüsse sind in etwa 10 mm Entfernung vom Gehäuse um 90° abzuwinkeln und die LEDs dann so zu verlöten, dass sie etwa 7 mm über der Platine stehen. Damit sind sie passend zu den Gehäusebohrungen montiert (siehe Abbildung 3).

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

Funktionskontrolle und Abgleich

Nach Abschluss des Aufbaus ist der USB-AD-Wandler mittels des mitgelieferten Testprogramms zu überprüfen. Dazu verbindet man den AD-Wandler über ein USB-Verbindungskabel mit einem PC. Dieser erkennt die neu angeschlossene Hardware und verlangt dann nach kurzer Zeit einen USB-Treiber. Dieser Treiber (ftd2xx.inf) befindet sich auf der mitgelieferten Programmdiskette.

Daraufhin ist das Testprogramm „usb_ad.exe“ von der mitgelieferten Programmdiskette zu starten, woraufhin das Dialogfeld (Abbildung 4) erscheint. Zuerst ist in diesem Fenster der Abgleich durch Betätigung der Taste „Abgleich“ zu starten. Es erscheint das Fenster des Max-Abgleichs (siehe Abbildung 5). An den

Kanal 1 des UAD 8 (Wannenstecker BU2: Pin 1 = 2,500 V, Pin 2 = GND) ist eine mit einem Multimeter eingestellte Spannung von 2,500 V anzulegen. Durch Betätigung des Buttons „OK“ gelangt man in den „Offset-Abgleich“. Jetzt ist an den Kanal 1 GND-Potenzial (Wannenstecker BU2: Brücke zwischen Pin 1 und Pin 2) anzulegen. Nach Betätigung der Taste „OK“ erscheint eine Meldung, dass der Abgleich erfolgreich beendet ist – ansonsten ist der gesamte Abgleich zu wiederholen.

Nach einem erfolgreichen Abgleich werden die Abgleichdaten im internen EEPROM des Mikrocontrollers gespeichert und bei Bedarf abgerufen. Diese Daten bleiben auch nach einem Spannungsausfall (USB-Verbindung unterbrochen) erhalten, so braucht der Abgleich nicht vor jeder Verwendung des UAD 8 erneut durchgeführt werden.

Mit dem Button „Konfig“ öffnet sich ein Dialogfenster „Konfiguration“ (siehe Abbildung 6), in dem man die Konfiguration

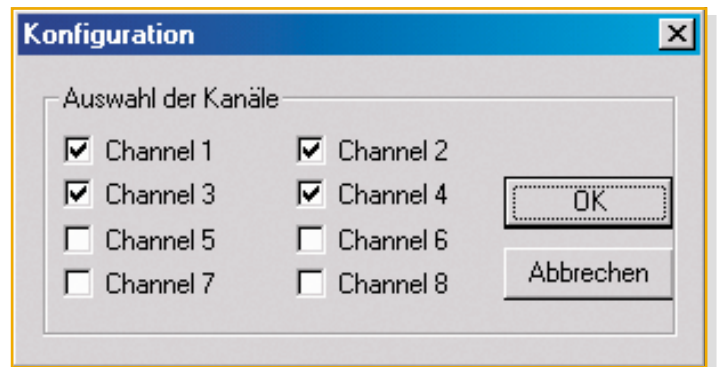


Bild 6: Das Konfigurationsmenü des Testprogramms

der verwendeten Kanäle vornimmt. Durch Setzen der Häkchen vor dem jeweiligen Kanal wird dieser aktiviert. In der Abbildung 6 sind beispielhaft die Kanäle 1 bis 4 aktiviert.

Unter dem Punkt „Aktiv“ wird das UAD-8-Testprogramm eingeschaltet. Der UAD 8 misst in den konfigurierten Kanälen, sobald die Betriebsspannung anliegt bzw. die USB-Verbindung zu einem „laufenden“ PC hergestellt ist. Das UAD-8-Testprogramm fragt die Messdaten aller Kanäle ab und zeigt die Werte im jeweiligen Fenster an. In dem Fenster eines nicht konfigurierten Kanals wird „-.-.-“ angezeigt.

Das Testprogramm fragt die Messdaten der AD-Wandler-Eingänge in einem festen Zeitraster ab und aktualisiert die Werte. Der Funktionstest ist beendet, wenn mit jedem AD-Wandler-Eingang die angelegte Spannung gemessen werden kann.

Nach diesen Tests kann der USB-AD-Wandler in eigene Applikationen integriert werden.

