



USB-DA-Wandler UDA 7

Der PC rückt im Elektroniklabor immer mehr in den Mittelpunkt des Equipments. Für seine Anbindung an die analoge Umwelt benötigt er jedoch immer wieder ein Interface – und das heute zunehmend über die weiter an Bedeutung gewinnende USB-Schnittstelle. Nach der Vorstellung des USB-AD-Wandlers UAD 8 im letzten „ELVjournal“ beschreiben wir hier das Pendant, das auf 8 Kanälen vom PC erzeugte Spannungen zwischen 0 und 2,55 V ausgibt.

Grenzen überwinden

Viele PC-Besitzer nutzen ihren PC nicht nur als Schreib- oder Spielgerät, sondern programmieren die unterschiedlichsten Anwendungen selbst. Da werden Internetseiten oder auch Datenbanken für den alltäglichen Gebrauch erstellt. Hierfür wird in der Regel keine spezielle Hardware benötigt, so dass der „Hobby-Programmierer“ sofort nach dem Starten des PCs mit seinen eigenen Projekten beginnen kann.

Etwas komplizierter wird es jedoch, wenn der PC mit externen Geräten, Messaufbauten und eigenen Elektronik-Applikationen kommunizieren soll. Denn hier beginnt meist die analoge Elektronikwelt, für die der PC (außer seiner Soundkarten-Option) von Haus aus keine Schnittstelle besitzt.

Dennoch – ein immer beliebter Bereich der Elektronik (gerade auch bei den Hobbyprogrammierern) stellt die Messwertaufnahme und -verarbeitung per PC dar. Hier werden unterschiedliche analoge

Datensignale via AD-Wandler aufgezeichnet und entsprechend ausgewertet. Aber auch der umgekehrte Weg einer DA-Wandlung stellt eine oft verwendete Anwendung dar. Sollen beispielsweise mit einem PC

Technische Daten: USB-DA-Wandler

Ausgänge:

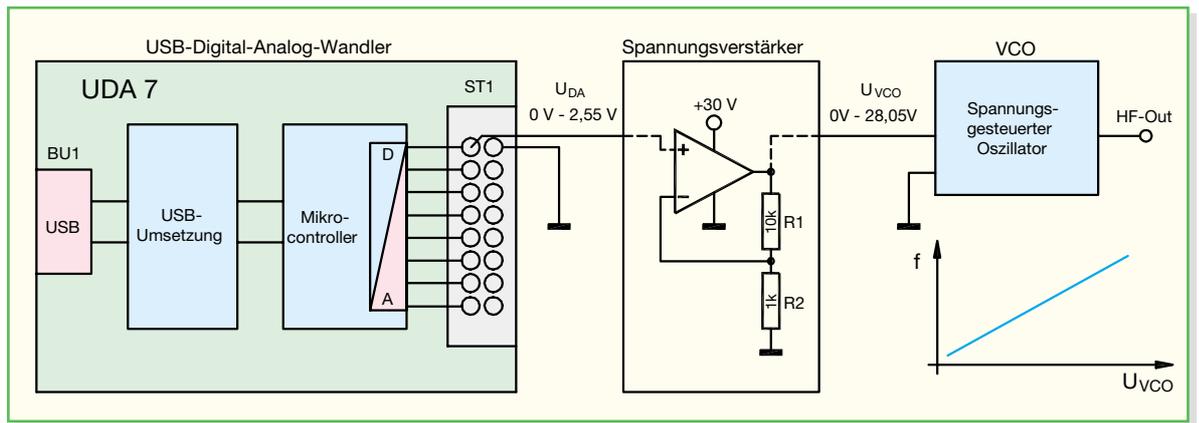
Anzahl: 8 unabhängige Kanäle
 Auflösung: 10 mV bzw. 8 Bit
 Max. Ausgangsspannung: 2,55 V
 Strombelastung: 5 mA
 Max. Länge der Anschlussleitung: 3 m

USB-Schnittstellen-Konfiguration: 19.200 Baud, 8 Datenbit, gerade Parität, 1Stoppbit

Allgemein:

Spannungsversorgung: über USB
 Max. Stromaufnahme: 30 mA (unbelastet)
 Gehäuseabm. (B x H x T): 92 x 24 x 69 mm
 Systemvoraussetzung: WIN 98, ME, XP, 2000, USB-Schnittstelle

Bild 1:
Anwendungs-
beispiel
des UDA 7



unterschiedliche Spannungen ausgegeben werden, so ist auch hier der Einsatz einer externen PC-Hardware – eines DA-Wandlers notwendig. Nun kann man einwenden, dass dies ein altes Thema ist, Digital-Analog-Wandler für die serielle Schnittstelle gibt es wie Sand am Meer. Aber gerade die früher als Standard erkorene serielle Schnittstelle in Form des 9-poligen RS-232-Anschlusses verschwindet zusehends zugunsten moderner, schnellerer Schnittstellen wie USB oder Firewire. Viele tragbare Computer verfügen gar nicht mehr über die „alte“ Schnittstelle. Deshalb müssen moderne Interfaces (z. B. AD- und DA-Wandler) auch für die modernen Schnittstellen ausgelegt sein. Diese tragen dann neben der Datenausgabe auch gleich die Intelligenz zur Kommunikation mit der (nicht einfach programmierbaren) USB-Schnittstelle mit sich.

Der USB-DA-Wandler UDA 7 bietet sich hierbei als Verbindungsglied von der digitalen zur analogen Außenwelt an. Mit ihm lassen sich digital in einem PC-Programm vorgegebene Werte in analoge Spannungen umsetzen. Der Anschluss an den PC wird über eine USB-Schnittstelle hergestellt. Der UDA 7 stellt acht DA-Wandler-Ausgänge zur Verfügung, an denen sich analoge Spannungen im Bereich von 0 V bis 2,55 V erzeugen lassen. Der DA-Wandler hat eine Auflösung von 8 Bit, wodurch sich bei gegebenem Spannungsbereich eine Schrittweite von 10 mV ergibt.

Der USB-DA-Wandler

Einsatz, Beschaltung und Anschluss

Bei den meisten Anwendungen eines DA-Wandlers geht es darum, die Steuerung einer Schaltung, eines elektronischen Gerätes etc. zu automatisieren bzw. eine manuelle Bedienung durch eine PC-(fern-)gesteuerte zu ersetzen. Ganz allgemein wird ein solcher DA-Wandler meist dort eingesetzt, wo analoge Eingaben bzw. Einstellungen durch digitale Eingabe bzw. Steuerung zu ersetzen sind. Das einfachste Bei-

spiel ist hier wohl ein (analoges) Potentiometer, dessen Funktion als Drehwinkelabhängige Steuerspannungsquelle durch einen DA-Wandler nachgebildet wird.

Hierzu lassen sich viele weitere Beispiele bringen, wobei die Variation des Aussteuerungsbereiches, d. h. ob die Ausgangsspannung einen Bereich von 0 V bis 2,55 V, bis 5 V, bis 10 V oder bis 30 V überstreichen muss, sehr vielfältig ist. Hier wird es sicherlich bei vielen Anwendungen erforderlich sein, eine entsprechende Anpassung in Form eines einfachen Spannungsverstärkers zu realisieren.

Der Aussteuerungsbereich des ELV UDA 7 von 0 V bis 2,55 V ergibt sich aus der Tatsache, dass die gesamte Schaltung aus der 5-V-Betriebsspannung der USB-Schnittstelle versorgt wird. Weiterhin ist es vielfach von Vorteil, eine einfach nachvollziehbare Schrittweite der analogen Ausgangsspannung zu haben. Hier bietet es sich bei einem 8-Bit-Wandler, der ja die Digitalwerte 0 bis 255 verarbeiten kann, an, den Bereich von 0 V bis 2,55 V zu definieren und so diesen dann in Schritten zu 10 mV durchlaufen zu können.

Als prinzipielle Anwendung für einen DA-Wandler ist beispielsweise in der HF-Technik die Steuerung von Verstärkern oder die Einstellung von Signaldämpfungen zu nennen. Beides kann benutzt werden, um einen HF-Pegel komfortabel einstellbar zu machen oder, in Verbindung mit einem entsprechenden AD-Wandler, eine Pegelregelung zu realisieren. Eine weitere Anwendung ist die Steuerung eines VCOs (voltage-controlled oscillator). Hier erfolgt die Frequenzeinstellung mit Hilfe einer analogen Steuerspannung.

Aber auch ganz andere Anwendungsbereiche sind denkbar: So lässt sich der DA-Wandler zur Steuerung von Modellbahnanlagen oder zur Drehzahl-einstellung bei einfachen DC-Motoren verwenden. Eine klassische Spannungsschnittstelle, die sehr elegant durch einen

DA-Wandler bedient werden kann, ist die „0–10-V-Analogschnittstelle“ in der Lichttechnik. Bei Dimmerracks werden so die Helligkeitswerte der angeschlossenen Leuchten mit Hilfe dieser Steuerspannung, die im Bereich von 0 V (dunkel) bis 10 V (max. Helligkeit) liegt, verändert.

Auch im Laborbereich gibt es diverse Anwendungen, sei es, um zum automatischen Testen von DC-Verstärkern eine variable Eingangsspannung zu generieren (und diese dann ggf. am Ausgang des Verstärkers mit Hilfe eines entsprechenden AD-Wandlers wieder einzulesen), oder um die Auswirkung bei der Variation einer Referenzspannung zu ermitteln.

Als konkretes Beispiel ist in Abbildung 1 die Vorgehensweise bei der Ansteuerung eines VCOs dargestellt: Der DA-Wandler liefert an seinem Ausgang eine Spannung im Bereich von 0 V bis 2,55 V. Diese reicht zur Ansteuerung eines VCOs, der üblicherweise mit Steuerspannungen von 1 V bis 28 V arbeitet, nicht aus. Daher erfolgt mit Hilfe des gezeichneten Pufferverstärkers eine Anpassung der Pegel. Im einfachsten hier gezeichneten Fall wird so der Spannungsbereich 0 V bis 2,55 V in den Bereich von 0 V bis 28,05 V gehoben. Der gezeichnete Pufferverstärker ist dabei ein einfacher Operationsverstärker, der als nicht-invertierender Verstärker geschaltet ist. Die Verstärkung bestimmt sich über folgende Formel:

$$U_{VCO} = U_{DA7} \cdot \left(1 + \frac{R1}{R2}\right)$$

Will man den Verstärker z. B. bis 28 V aussteuern, so lässt sich das dafür benötigte Widerstandsverhältnis über die umge-

Tabelle 1: Verwendete Steuerzeichen

Zeichen	Wert	Bedeutung
STX	0x02	//Start of Text
ETX	0x03	//End of Text
ENQ	0x05	//Enquire
ACK	0x06	//Acknowledge „OK“
NAK	0x15	//not Acknowledge „nicht OK“
DC2	0x12	
DC3	0x13	

Tabelle 2: Befehlssatz des UDA 7

Bedeutung	Befehl	Parameter	Antwort	Bedeutung
Einstellung CH 1	'1' (31hex)	Ausgangsspannung (0x00-0xFF)	ACK NAK	OK nicht OK
Einstellung CH 2	'2' (32hex)	Ausgangsspannung (0x00-0xFF)	ACK NAK	OK nicht OK
Einstellung CH 3	'3' (33hex)	Ausgangsspannung (0x00-0xFF)	ACK NAK	OK nicht OK
Einstellung CH 4	'4' (34hex)	Ausgangsspannung (0x00-0xFF)	ACK NAK	OK nicht OK
Einstellung CH 5	'5' (35hex)	Ausgangsspannung (0x00-0xFF)	ACK NAK	OK nicht OK
Einstellung CH 6	'6' (36hex)	Ausgangsspannung (0x00-0xFF)	ACK NAK	OK nicht OK
Einstellung CH 7	'7' (37hex)	Ausgangsspannung (0x00-0xFF)	ACK NAK	OK nicht OK
Einstellung CH 8	'8' (38hex)	Ausgangsspannung (0x00-0xFF)	ACK NAK	OK nicht OK

stellte Formel einfach bestimmen:

$$\frac{R1}{R2} = \frac{U_{VCO}}{U_{DA7}} - 1$$

Bei den gegebenen Werten ergibt sich ein Verhältnis von 10:1. Damit lassen sich die Widerstände, wie im Beispiel gezeigt, ganz leicht dimensionieren. So kann dann bequem über den PC die Ausgangsfrequenz des Oszillators variiert werden. Diese Vorgehensweise lässt sich prinzipiell auf alle praktischen Anwendungsbeispiele übertragen.

Der Anschluss des DA-Wandlers UDA 7 ist relativ simpel. Einzig die Verbindung zum PC ist nötig. Daten und Spannungsversorgung kommen dann über die USB-Schnittstelle vom PC. Die analogen Ausgangssignale stehen dann an dem 16-poligen Wannenstecker zur Verfügung.

Datenübertragung

Die USB-Datenübertragung erfolgt hier – ähnlich wie bei einer seriellen Daten-

übertragung mittels RS-232-Schnittstelle – mit einer Baudrate von 19.200 Bit/s. Wobei als Parameter 8 Datenbit, gerade Parität und 1 Stoppbit gewählt wurde. Eine Erläuterung dieses Datenübertragungsprotokolls ist in der Infobox „Serielle Datenübertragung“ zu finden.

Dies stellt jedoch erst die untere, allgemeine Ebene des Datenübertragungsprotokolls dar, das verwendete Protokoll des UDA 7 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 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 verhin-

dern. D. h. diese gemäß der Tabelle 1 dargestellten Spannungswerte können nicht in ihrer ursprünglichen Form übertragen werden, da der Empfänger diese als Steuerzeichen interpretiert und nicht als Spannungswerte. Daher wird eine Konvertierung der Zeichen (Werte) entsprechend folgender Auflistung durchgeführt (eine Übersicht der verwendeten ASCII-Zeichen befindet sich in Tabelle 2):

```
<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 zu ersetzen. Empfängerseitig wird diese Umsetzung wieder rückgängig gemacht, um die eigentlichen Nutzdaten wiederherzustellen. Das Übertragungsbeispiel verdeutlicht dies nochmals. Der entsprechende Befehlssatz des UDA 7 ist in Tabelle 2 zu sehen und dort erläutert.

Übertragungsbeispiel

Wir wollen das verwendete Protokoll einmal an einem Beispiel betrachten. Als Beispiel für den Umgang mit dem Befehlssatz soll an Kanal 6 eine Spannung von 2,36 V (entsprechender digitaler Wert: 0xEChex = 236dec) und an Kanal 3 eine Spannung von 20 mV (Digitalwert 0x02hex = 2dec) eingestellt werden.

Die Auswahl des Kanals 6 (siehe Tabelle 2) wird mit dem Befehl „6“ (36hex) vorgenommen. Der folgende Parameter gibt die gewünschte Ausgangsspannung als Hex-Wert (0xEChex) an. Die gesamte Befehlszeile setzt sich wie folgt zusammen:

```
<STX> <36hex> <EChex> <ETX>
```

Die Auswahl des Kanals 3 wird mit dem

Infobox „Serielle Datenübertragung“

Die Geschwindigkeit der Datenübertragung wird üblicherweise in „Bit/s“ angegeben. Diese Angabe kennzeichnet die Anzahl von einzelnen Bit, die pro Sekunde übertragen werden. Jedes übertragene Bit ist zusätzlich in einen entsprechenden Datenrahmen gefasst, der beim UDA 7 aus drei zusätzlichen Bit besteht. Der Datenrahmen umrahmt durch ein Start- und ein Stoppbit die Kombination aus Daten und Paritätsbit (siehe unten). Er dient dazu, dass der Empfänger den Beginn eines Datenblocks eindeutig erkennt und dann die übertragenen Bit entsprechend erfasst. Jeder Datenblock beginnt mit einer logischen „1“ als Startbit und endet mit einer logischen „0“ als

Stoppbit. Die Anzahl der Stoppbits ist beim UDA 7 auf eines festgelegt, jedoch können andere Geräte auch mit 1,5 oder 2 Stoppbit arbeiten. Es werden also nicht nur die reinen „Nutzdaten“ übertragen. Aus diesem Grunde ist die effektive übertragene Anzahl an Daten natürlich etwas geringer, als es die reine Angabe der Datenübertragungsgeschwindigkeit aussagt.

Bei jeder Datenübertragung können Fehler auftreten, die durch verschiedene Fehlererkennungsmethoden vermittelt werden können. Eine einfache und effektive Methode ist das Einfügen eines Paritätsbit, bei dem zwischen gerader (even parity) und ungerader Parität (odd parity) unterschieden wird. Der Sender untersucht hier das

zu übertragene 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 zurückgesetzt, 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. Dies ist für einfache Datenübertragungen, wie beim UDA 7, ausreichend.

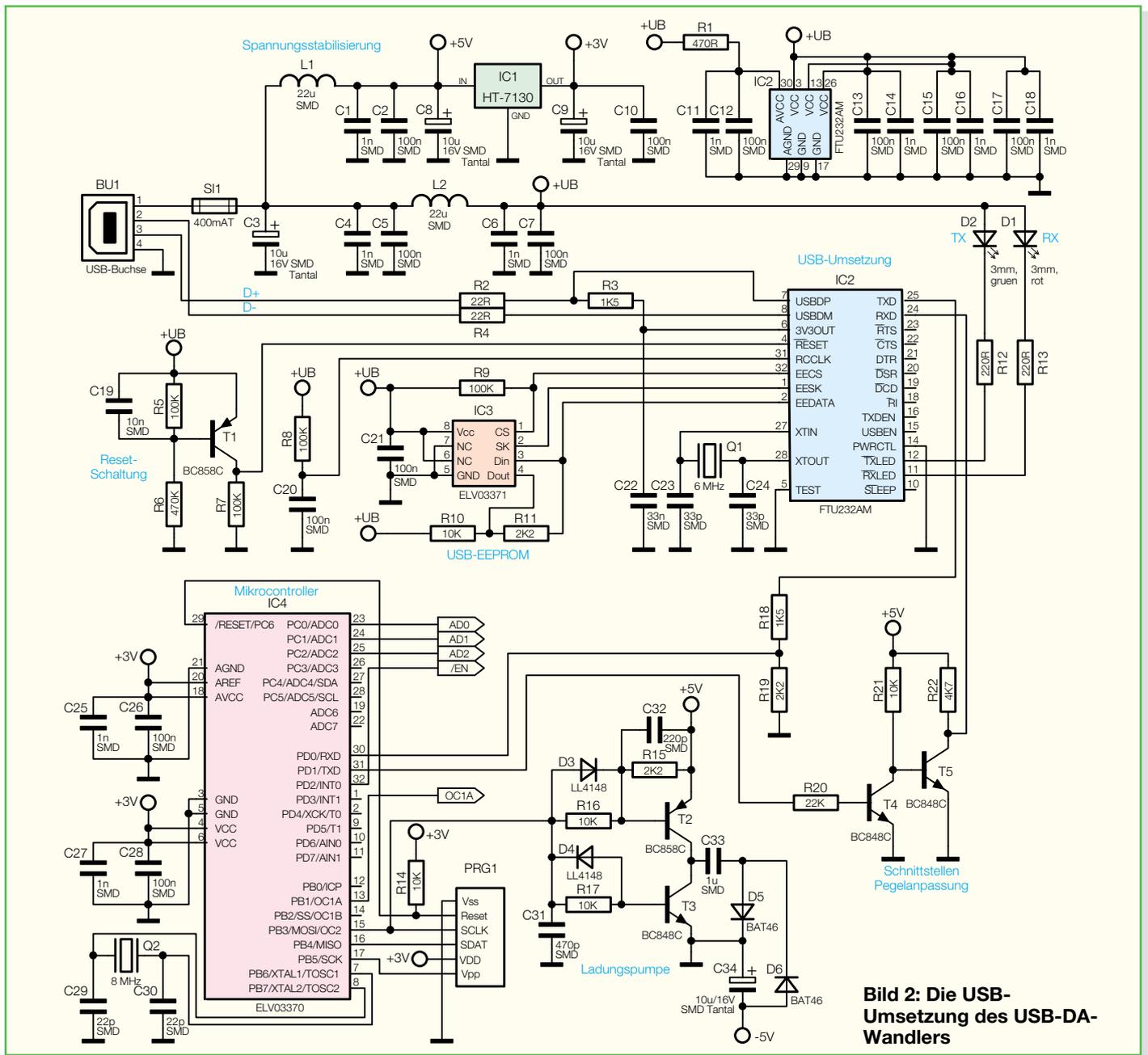


Bild 2: Die USB-Umsetzung des USB-DA-Wandlers

Befehl „3“ (33hex) vorgenommen. Über den zweiten Parameter wird der gewünschte Spannungswert (0x02hex) eingestellt. Der Befehl setzt sich dann wie folgt zusammen.

```
<STX> <33hex> <02hex> <ETX>
```

Hier ist jedoch bereits zu sehen, dass der Befehlsparameter das Steuerzeichen „STX“ (0x02hex) enthält. Es muss also folgende Konvertierung erfolgen:

```
<STX> <33hex> <ENQ> <DC2> <ETX>
```

Diese zusammengesetzten Datenstrings sind über die USB-Schnittstelle an die angeschlossene UDA-7-Hardware zu übertragen. Dort werden die jeweiligen digitalen Spannungswerte vom Mikrocontroller an

den eigentlichen DA-Wandler weitergeben und als entsprechende Analogspannung ausgegeben. Außerdem speichert der Controller die Daten in seinem EEPROM ab, so dass nach einem Reset der UDA-7-Hardware die vorher eingestellte Spannung wieder eingestellt wird. Weiterhin wird jeder richtig empfangene Datenstring mit einem „acknowledge“

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

quittiert. Ein fehlerhaft empfangener Datenstring wird dagegen mit einem „not acknowledge“

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

quittiert. Hat man diese Bildungsregeln erst einmal verinnerlicht, sollte die Programmie-

rung der einzelnen Befehle kein größeres Problem mehr darstellen.

Zum besseren Verständnis der Ansteuerung des USB-DA-Wandlers befindet sich auf der dem Bausatz beiliegenden Programmierscheibe ein Beispielprojekt mit zugehörigen Quelltexten in Visual C++. Außerdem findet man hier zahlreiche 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 UDA 7 ist zur besseren Übersicht in zwei Schaltbildern dargestellt. Beginnen wir mit dem Steuerteil (siehe Abbildung 2), und hier

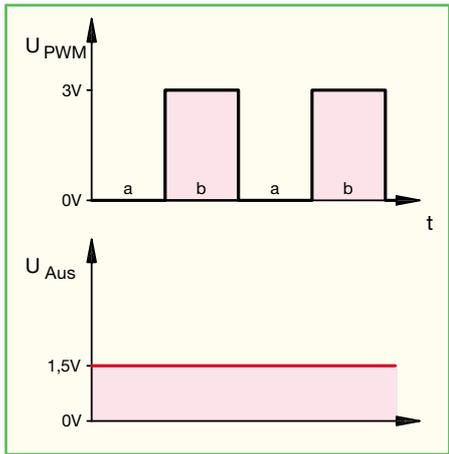


Bild 3: Das angewendete Prinzip der DA-Wandlung

mit der genaueren Beschreibung der USB-Umsetzung in der oberen Hälfte des Schaltbildes. Über die Buchse BU 1 gelangen die USB-Signale in die Schaltung. Der USB (Universal Serial Bus) stellt neben den beiden Datenleitungen (D+, D-) – dazu später mehr – auch eine Betriebsspannung von 5 V DC zur Verfügung, wobei die Sicherung SI 1 diese USB-Spannung gegen Überlastungen absichert. Die Kondensatoren C 1 bis C 8 sowie die Induktivitäten L 1 und L 2 dienen zur Stabilisierung der USB-Spannung bzw. zur Unterdrückung hochfrequenter Störungen. Der Spannungsregler IC 1 in Verbindung mit den Kondensatoren C 9 und C 10 stabilisiert die USB-Spannung auf 3,0 V, die zum Betrieb des Mikrocontrollers IC 4 benötigt wird. Die weiteren Kondensatoren (C 11 bis C 18) dienen zur weiteren Störunterdrückung.

Der USB-Controller IC 2 bildet das zentrale Element der USB-Umsetzung, da er speziell für die Konvertierung zwischen USB und RS 232 entwickelt wurde. Er setzt ankommende Befehle vom USB-Format in serielle RS-232-Befehle und abgehende serielle RS-232-Befehle in das USB-Format um, so dass der Mikrocontroller diese an seinen Schnittstellen-Pins (RXD und TXD) lesen und ausgeben kann. Da der USB-Controller IC 2 und der Mikrocontroller IC 4 mit unterschiedlicher Betriebsspannung arbeiten, ist hier eine Pegelanpassung vorzunehmen. Die Pegel der empfangenen Daten werden für den Mikrocontroller von 5 V auf 3 V herabgesetzt, dies erfolgt über einen einfachen Spannungsteiler (R 18, R 19). In der Gegenrichtung

werden die Pegel der gesendeten Daten mit Hilfe zweier einfacher Transistorstufen (T 4 und T 5 mit Beschaltung) in Emitterschaltung von 3 V auf 5 V heraufgesetzt. Die Taktung des USB-Controllers übernimmt ein interner Oszillator, der mit dem Quarz Q 1 und den Kondensatoren C 23 und C 24 auf eine Frequenz von 6 MHz stabilisiert wird. Ein definiertes Starten des Controllers wird durch Beschalten des Reset-Pins mit der Reset-Schaltung, bestehend aus T 1, R 5 bis R 7 und C 19, sichergestellt. Zur Speicherung der USB-Erkennungsdaten (Vendor-ID, Product-ID, Seriennummer etc.) des UDA 7 ist an die „Microwire“-Schnittstelle von IC 2 ein EEPROM (IC 3) angeschlossen.

Neben der USB-Spannung, wie beschrieben, liegen die beiden Datenleitungen (D+, D-) an BU 1 (Pin 2 und 3) an. Diese sind jeweils über einen Widerstand (R 2, R 4), der die Leitungsanpassung verbessert, mit dem USB-Controller IC 2 verbunden. Der Widerstand R 3 dient als Pull-up-Widerstand des USB. Über den definierten „High“-Pegel der D+-Datenleitung erkennt der PC eine angeschlossene UDA-7-Hardware. Die Leuchtdioden D1 und D2 fun-

tionen die Pegel der gesendeten Daten mit Hilfe zweier einfacher Transistorstufen (T 4 und T 5 mit Beschaltung) in Emitterschaltung von 3 V auf 5 V heraufgesetzt.

Die Taktung des USB-Controllers übernimmt ein interner Oszillator, der mit dem Quarz Q 1 und den Kondensatoren C 23 und C 24 auf eine Frequenz von 6 MHz stabilisiert wird. Ein definiertes Starten des Controllers wird durch Beschalten des Reset-Pins mit der Reset-Schaltung, bestehend aus T 1, R 5 bis R 7 und C 19, sichergestellt. Zur Speicherung der USB-Erkennungsdaten (Vendor-ID, Product-ID, Seriennummer etc.) des UDA 7 ist an die „Microwire“-Schnittstelle von IC 2 ein EEPROM (IC 3) angeschlossen.

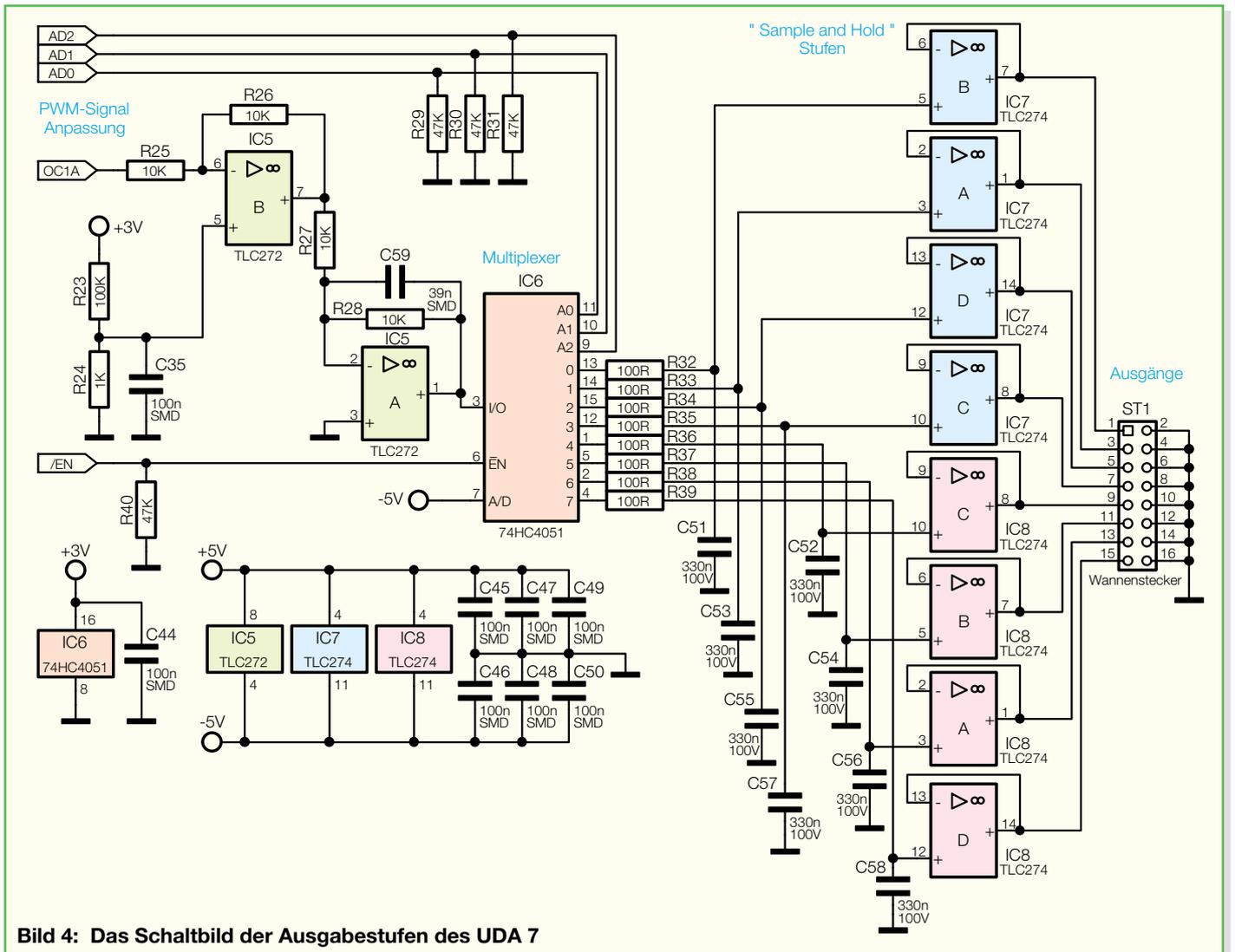


Bild 4: Das Schaltbild der Ausgabestufen des UDA 7

gieren als Statusanzeige für die Übertragung – die grüne LED (D 2) zeigt einen Sendevorgang und die rote LED (D 1) einen Datenempfang an.

In der unteren Hälfte des Schaltbildes bildet der Mikrocontroller IC 4 das zentrale Element. Dieser wandelt den gewünschten digitalen Spannungswert in PWM-Signale (Pin 13, „OC1A“) um, aus denen später die analoge Spannung erzeugt wird. Als Steuerleitungen für diese DA-Wandlung dienen die Signale „AD 0“ bis „AD 2“ und „/EN“. Weiterhin speichert der Mikrocontroller die Daten der jeweilig übermittelten Parameter in seinem internen EEPROM ab.

Die Kondensatoren C 25 bis C 28 bewirken eine weitere Störunterdrückung der Betriebsspannung des Mikrocontrollers. Der Quarz Q 2 und die Kondensatoren C 29 und C 30 stabilisieren den Takt des Mikrocontrollers auf 8 MHz. Da für den Betrieb der Operationsverstärker eine negative Betriebsspannung benötigt wird, ist eine Ladungspumpe implementiert. Dieser Schaltungsteil ist mit den Transistoren T 2 und T 3, den Widerständen R 15 bis R 17, den Dioden D 3 bis D 6 und den Kondensatoren C 31 bis C 34 aufgebaut. Die Ansteuerung erfolgt vom Mikrocontroller her, der ein Rechtecksignal auf die Ladungspumpe gibt. Hiermit werden dann die Transistoren T 2 und T 3 wechselnd durchgeschaltet und der Kondensator C 33 entsprechend (um-)geladen. Letztlich lässt sich dann hinter D 6, d. h. am Kondensator C 34, eine negative Spannung („-5 V“) abgreifen.

Der Programmieradapter PRG 1 und der Widerstand R 14 ermöglichen eine Programmierung des Mikrocontrollers in der Serienproduktion.

Die Hauptaufgabe des Mikrocontrollers ist die DA-Wandlung. In Abbildung 3 ist das Prinzip der hier angewendeten Wandlung kurz dargestellt. Der Mikrocontroller gibt ein Rechtecksignal (UPWM) mit den beiden Pegelwerten 0 V und +3 V aus. In dem hier gewählten Beispiel ist dabei ein Tastverhältnis von 1:1 gewählt, d. h. der „High“-Anteil ist genauso lang wie der „Low“-Anteil. Bildet man jetzt von diesem digitalen Signal den arithmetischen Mittelwert (z. B. mit Hilfe einer RC-Schaltung), so ergibt sich ein (in diesem Fall mittlerer) Spannungswert von 1,5 V. Durch das Variieren des Tastverhältnisses lassen sich so unterschiedliche Mittelwerte bilden, aus der PWM wird eine analoge Spannung. So weit zur Theorie – die praktische Schaltung dazu ist in Abbildung 4 zu sehen.

Das vom Mikrocontroller erzeugte PWM-Signal („OC1A“) gelangt auf den als invertierenden Verstärker geschalteten Operationsverstärker IC 5 B. Die Signalverläufe zu der folgenden weiteren Beschreibung sind in Abbildung 5 zu sehen.

Der invertierende Verstärker hat, bedingt durch die Beschaltung mit $R_{25} = R_{26}$, eine Verstärkung von $v_u = -1$. Somit müsste am Ausgang des OPV das Eingangssignal genau invertiert anstehen. Über eine kleine Vorspannung am positiven Eingang (Pin 5) von IC 5 B, die mit Hilfe des Spannungsteilers aus R 23 und R 24 entsteht, kommt es zu einer gewollten Verschiebung des Ausgangssignals. Die Ausgangsspannung ist um ca. 30 mV ins Positive verschoben. Der mittlere Graph in Bild 5 zeigt hier sehr schön, dass IC 5 B das Eingangssignal invertiert und um 30 mV nach oben verschiebt. Dieser Trick ist notwendig, um den Ausgangsspannungsbereich des UDA 7 bei 0 V beginnen zu können. Der durch die Operationsverstärker entstandene künstliche Offset wird später beim Abgleich durch den Mikrocontroller kompensiert, so dass beim gewünschten digitalen Ausgangswert 0 auch 0,00 V am Ausgang anstehen.

Das so angepasste PWM-Signal gelangt auf den nachgeschalteten Operationsverstärker IC 5 A. Dieser ist ebenfalls, bedingt durch die Beschaltung mit $R_{27} = R_{28}$, als invertierender Verstärker mit der Verstärkung $v_u = -1$ geschaltet. Der zu R 28 parallel geschaltete Kondensator C 59 wirkt als Integrationsglied. Er sorgt dafür, dass sich am Ausgang des Operationsverstärkers nach einer definierten Ladezeit der arithmetische Mittelwert des PWM-Signals einstellt. Die Höhe der Spannung U_x ergibt sich dabei aus dem angelegten Tastverhältnis des PWM-Signals. Am Ausgang von IC 5 A liegt somit das digital-analog-gewandelte Ausgangssignal an. Da der USB-DA-Wandler aber nicht nur einen analogen Ausgang hat, sondern an insgesamt 8 Ausgängen entsprechende Analogspannungen ausgeben kann, ist eine „sample-and-hold“-Schaltung erforderlich.

Prinzipiell stellt eine solche Schaltung eine Zwischenspeicherung der einzelnen, insgesamt acht Ausgangsspannungen dar. Im ersten Schritt erfolgt die DA-Wandlung des entsprechenden analogen Spannungswertes. Anschließend schaltet ein Multiplexer diesen Spannungswert kurzzeitig auf einen dem entsprechenden Kanal zugehörigen Speicherkondensator. Während der DA-Wandler die Werte für die nächsten Kanäle wandelt, kann die Spannung am Speicherkondensator, über einen Pufferverstärker entkoppelt, am Ausgang abgegriffen werden.

Der genaue Ablauf stellt sich wie folgt dar: Der Mikrocontroller schaltet den Multiplexer (IC 6) über den „/EN“-Anschluss inaktiv. Anschließend stellt dieser das gewünschte PWM-Signal für Kanal 1 an seinem PWM-Ausgang („OC1A“) ein. Dadurch stellt sich am Ausgang des Operationsverstärkers IC 5 A eine dem Mittelwert

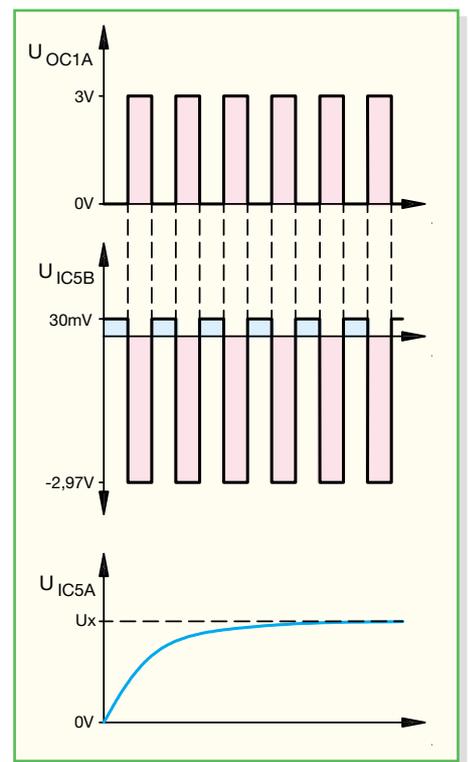
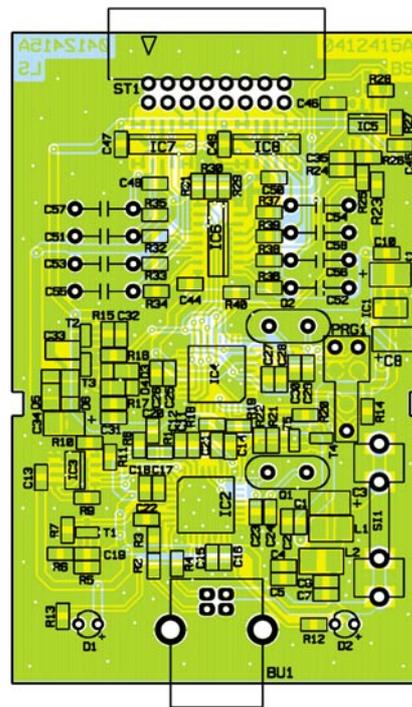
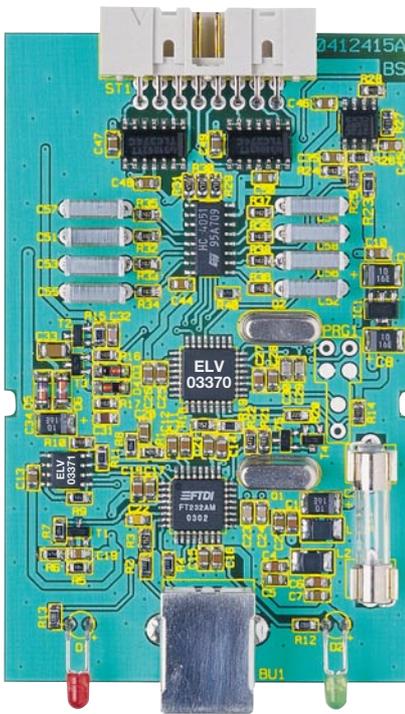


Bild 5: Die Aufbereitung des vom Mikrocontroller erzeugten PWM-Signals

des PWM-Signals entsprechende Spannung ein. Damit dieser Ladevorgang korrekt ablaufen kann, wartet der Mikrocontroller eine entsprechende Zeit. Anschließend wird der Kanal 1 über die Adressleitungen „AD 0“ bis „AD 2“ ausgewählt und mit der Freigabe durch die „/EN“-Leitung aktiviert. Damit lädt die Ausgangsspannung des DA-Wandlers IC 5 A den Sample-Hold-Kondensator für Kanal 1 (C 51) auf. Über den nachgeschalteten Operationsverstärker (IC 7 B), der als Impedanzwandler geschaltet ist, steht diese Spannung sofort am Ausgang ST 1, Pin 1 des UDA 7, an. Dieser Impedanzwandler ist notwendig, um den Sample-Hold-Kondensator nur extrem hochohmig zu belasten und gleichzeitig die Ausgangsspannung niederohmig zur Verfügung zu stellen.

Nach kurzer Umladungszeit schaltet der Mikrocontroller den Multiplexer wieder inaktiv („/EN“ = high) und stellt das gewünschte PWM-Signal für Kanal 2 an seinem PWM-Ausgang („OC1A“) an. Nach kurzer Wandelzeit steht am Ausgang des DA-Wandlers IC 5 A die Spannung für Kanal 2 an. Am Multiplexer wird über die Adress-Leitungen „AD 0“ bis „AD 2“ der Kanal 2 eingestellt und mit „/EN“ aktiviert. Nun erfolgt ein Laden des Kondensators für Kanal 2 (C 53). Über den nachgeschalteten Impedanzwandler IC 7 A lässt sich diese Spannung am Ausgang ST 1, Pin 3, abgreifen. Dieser Vorgang verläuft in einer Endlosschleife, so dass nach Kanal 2



Ansicht der fertig bestückten Platine des USB-DA-Wandlers mit zugehörigem Bestückungsplan

der Kanal 3 usw. und nach Kanal 8 der Kanal 1 folgt. Durch dieses zyklische Nachladen der Kondensatoren stellt sich an den Ausgängen eine stabile, der Spannungsstufe entsprechende Spannung ein. Bei der Beschaltung der Ausgänge ist zu beachten, dass der maximale Ausgangsstrom der Operationsverstärker IC 7 und IC 8 eingehalten wird (siehe technische Daten).

Nachbau

Der UDA 7 ist aus der Zielstellung, eine besonders kompakte Baugruppe zu entwerfen, fast ausschließlich mit SMD-Bauteilen aufgebaut. Ein geregelter LötKolben mit sehr feiner Spitze, SMD-Lötzinn, Entlötlitze und eine SMD-Pinzette zum Positionieren der kleinen Bauteile zählen zur Grundausstattung für den Aufbau.

Der Aufbau erfolgt anhand des Bestückungsdrucks, des Platinenfotos sowie der Stückliste. Die Bestückung beginnt mit dem Auflöten von IC 1 bis IC 8. Diese haben einen geringen Pin-Abstand und sind am einfachsten zu bestücken, wenn

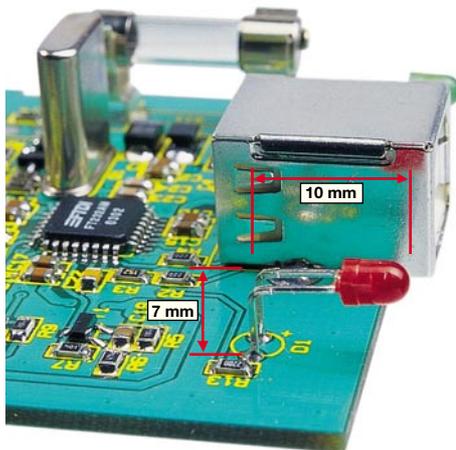


Bild 6: So erfolgt die LED-Bestückung.

ringsum noch keine Bauteile die Lötarbeiten behindern. Wichtig ist hier die korrekte Einbaulage der ICs. Beim USB-Controller (IC 2) und dem Mikrocontroller (IC 4) ist die Pin 1 zugeordnete Ecke entweder angeschragt oder durch eine kreisförmige Ausfräsung des Gehäuses gekennzeichnet, die sich auch im Bestückungsdruck wiederfindet. Bei den restlichen ICs ist die Pin 1 zugeordnete Seite abgeflacht bzw. durch eine Gehäusekerbe gekennzeichnet. Beim Spannungsregler IC 1 ergibt sich die polrichtige Bestückung automatisch aus der Pinkonfiguration und der Anordnung der zugehörigen Löt pads.

Man verzinnt zunächst ein Löt pad für das jeweilige IC vor, setzt dann das IC lagerichtig auf die Löt pads auf und verlötet den ersten Pin. Im Anschluss daran ist ein zweiter Pin an der diagonal gegenüberliegenden Seite zu verlöten. Dabei ist darauf zu achten, dass alle Anschlüsse des jeweiligen ICs genau auf den zugehörigen Löt pads aufliegen, um später Kontaktfehler durch ungenügende Verlötung oder gar Kurzschluss auszuschließen. Wichtig ist bei diesem ersten Schritt auch, dass alle Pins auf der Platine aufliegen. Wenn Pins „in der Luft schweben“, kann es beim späteren Verlöten dazu kommen, dass kein Kontakt zwischen Pin und Löt pad entsteht. Bevor die weiteren Anschlüsse der ICs mit der Leiterplatte verlötet werden, ist nochmals die richtige Position zu überprüfen, da eine Korrektur im Nachhinein nur noch sehr schwer durchführbar ist.

Nach dem Verlöten der IC-Pins und der Kontrolle der Lötstellen geht es nun an die weiteren SMD-Komponenten – die SMD-Widerstände, -Spulen, -Kondensatoren, -Transistoren und -Dioden. Hier wird ebenfalls zunächst jeweils ein Löt pad auf der Leiterplatte vorverzinnt, bevor man das

Stückliste: USB-DA-Wandler UDA 7

Widerstände:

22 Ω/SMD	R2, R4
100 Ω/SMD	R32–R39
220 Ω/SMD	R12, R13
470 Ω/SMD	R1
1 kΩ/SMD	R24
1,5 kΩ/SMD	R3, R18
2,2 kΩ/SMD	R11, R15, R19
4,7 kΩ/SMD	R22
10 kΩ/SMD	R10, R14, R16, R17, R21, R25–R28
22 kΩ/SMD	R20
47 kΩ/SMD	R29–R31, R40
100 kΩ/SMD	R5, R7–R9, R23
470 kΩ/SMD	R6

Kondensatoren:

22 pF/SMD	C29, C30
33 pF/SMD	C23, C24
220 pF/SMD	C32
470 pF/SMD	C31
1 nF/SMD	C1, C4, C6, C11, C14, C16, C18, C25, C27
10 nF/SMD	C19
33 nF/SMD	C22
39 nF/SMD	C59
100 nF/SMD	C2, C5, C7, C10, C12, C13, C15, C17, C20, C21, C26, C28, C35, C44–C50
330 nF/100 V	C51–C58
1 µF/SMD/Bauform 1206	C33
10 µF/16 V/SMD	C3, C8, C9, C34

Halbleiter:

HT7130/SMD	IC1
FT U232AM/SMD	IC2
ELV03371/SMD	IC3
ELV03370/SMD	IC4
TLC272/SMD	IC5
74HC4051/SMD	IC6
TLC274C/SMD	IC7, IC8
BC858C	T1, T2
BC848C	T3–T5
LL4148	D3, D4
BAT46/SMD	D5, D6
LED, 3 mm, rot	D1
LED, 3 mm, grün	D2

Sonstiges:

Quarz, 6 MHz, HC49U	Q1
Quarz, 8 MHz, HC49U	Q2
SMD-Induktivität, 22 µH	L1, L2
USB-B-Buchse, winkelprint	BU1
Wannen-Steckleiste, winkelprint, 2 x 8-polig	ST1
Sicherung, 0,4 A, träge	SI1
Platinensicherungshalter (2 Hälften), print	SI1

Bauteil mit der Pinzette erfasst, positioniert und am vorverzinnten Pad anlötet. Nach der Kontrolle der korrekten Position des Bauteils sind die restlichen Anschlüsse 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. Die SMD-Elkos C 3, C 8, C 9 und C 34 tragen üblicherweise an ihrem Pluspol eine Markierung, diese muss mit jener im Bestückungsdruck übereinstimmen.

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 Pinkonfiguration ergibt. Die Dioden sind durch eine Ring-Markierung an der Kathode gekennzeichnet.

Vor der weiteren Bestückung sind alle SMD-Lötstellen sorgfältig zu kontrollieren, ggf. unter Zuhilfenahme einer Lupe. Sind alle SMD-Komponenten aufgelötet und überprüft, beginnt die Bestückung der bedrahteten Bauelemente. Die Quarze (Q 1, Q 2) und die Folienkondensatoren (C 51 bis C 58) sind gemäß Bestückungsdruck zu bestücken und zu verlöten. Die USB-Buchse BU 1, der Wannenstecker ST 1 und der Sicherungshalter SI 1 müssen direkt auf der Leiterplatte aufliegen, bevor die Anschlüsse verlötet werden, da die Lötstellen sonst bei späterer mechanischer Belastung relativ stark beansprucht werden.

Nun erfolgt die polrichtige Bestückung der Leuchtdioden, wobei die Anode durch den längeren Anschlusspin gekennzeichnet ist. Die LED-Anschlüsse sind in ca. 10 mm Abstand von der LED-Spitze um 90° abzuwinkeln. Danach sind die LEDs so zu verlöten, dass sich ein Abstand von etwa 7 mm zwischen LED und Platine ergibt (siehe Abbildung 6). Die so montierten LEDs passen dann genau in die

Bild 7: Die Montage der Platine ins Gehäuse



entsprechenden Bohrungen des optional erhältlichen Gehäuses.

Abschließend, nach nochmaliger Überprüfung der Leiterplatte auf Bestückungsfehler und Lötzinnbrücken, ist das passend für diese Baugruppe erhältliche Gehäuse zu montieren. Hierzu wird die Platine mit der USB-Buchse und den LEDs voran in die Gehäuseoberseite eingelegt, so dass die Platine in die „Gehäuse-Nasen“ einrastet (siehe Abbildung 7) und LEDs sowie USB-Buchse in den zugehörigen Aussparungen stehen. Abschließend ist die Gehäuseunterschale von der Seite her aufzuschieben. Das so fertig aufgebaute Gerät ist nun noch zu testen und abzugleichen.

Funktionskontrolle und Abgleich

Für den Abgleich ist ein genaues Multimeter (besser 0,3 % Genauigkeit im DC-Messbereich bis 3,0 V). Außerdem wird ein PC mit USB-Schnittstelle und das mitgelieferte Testprogramm benötigt.

Das Testprogramm macht die Überprüfung der Funktion des USB-DA-Wandlers einfach. Dazu verbindet man zunächst die UDA-7-Hardware über ein USB-Anschlusskabel mit dem PC. Dieser erkennt die neu angeschlossene Hardware und verlangt nach kurzer Zeit einen USB-Treiber. Dieser (ftd2xx.inf) ist auf der mitgelieferten Programmdiskette enthalten. Anschließend ist den Anweisungen des Betriebssystems zu folgen, bis die Hardware korrekt installiert ist.

Nach dieser Installation ist das Testprogramm „usb_da.exe“ von der Programmdiskette zu starten, worauf das Dialogfeld des Testprogramms (siehe Abbildung 8) erscheint. Hier beginnt nun der eigentliche Abgleich:

Am Ausgang des Kanals 1 (ST 1, Pin 1 „+“ und Pin 2 „GND“) muss das Multimeter angeschlossen werden. Dann ist im Programmfenster der 2,55-V-Abgleich durch Setzen des Häkchens vor „2,55-V-Abgleich“ zu starten. (**Hinweis:** Unbedingt mit dem 2,55-V-Abgleich beginnen.) Es erscheint eine Meldung, dass die Ausgangsspannung an Channel 1 zu messen und mit den „Wipptasten“ des Kanal-Abgleichfeldes „Channel 1“ einzustellen ist (siehe Abbildung 9). Hier ist der am Bildschirm dargestellte Wert so zu verändern, dass sich am Multimeter eine Spannung von 2,55 V ergibt.

Ist dies geschehen, so ist das Häkchen vor 2,55-V-Abgleich wieder durch Anklicken zu löschen. Anschließend ist der Offset-Abgleich zu starten, indem das Häkchen vor das „0-V-Abgleich“-Feld gesetzt wird. Die Spannung am Ausgang von Kanal 1 ist nun über die „Wipptasten“ „Channel 1“ in gleicher Weise auf 0,00 V (am Multimeter) einzustellen. Nach erfolgreichem Abgleich des Offsets ist das Häkchen auch hier wieder zu löschen. Diese Abgleichdaten sind dann im internen EEPROM des Mikrocontrollers gespeichert und werden bei jeder Ausgabe entsprechend als Korrekturwert herangezogen. Diese Daten bleiben auch nach einem Spannungsausfall (USB-Verbindung unterbrochen) erhalten, so dass man den Abgleich nicht vor jeder Verwendung des UDA 7 erneut durchführen muss.

Nach diesen einfachen Abgleicharbeiten ist der USB-DA-Wandler einsatzbereit und kann wie vorgesehen genutzt werden. Das Testprogramm ist auch als einfaches Bedienprogramm für den UDA 7 einsetzbar. Dabei dürfen zunächst keine Häkchen in den beiden Abgleichoptionen gesetzt sein. Mit den jeweiligen Wipptasten des anzusprechenden Kanals lässt sich die Ausgangsspannung für diesen Kanal vorgeben. Sind alle gewünschten Kanäle eingestellt, schickt man die Daten durch Betätigen des „Aktualisieren“-Buttons zum UDA 7, an dessen Ausgängen dann die eingestellten Werte erscheinen. Damit findet der interessierte Programmierer bereits in diesem einfachen Programm einen Ansatz für eigene Applikationen. **ELV**

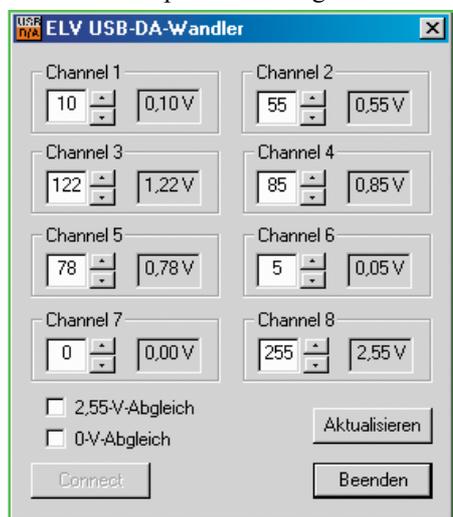


Bild 8: Mitgeliefertes Testprogramm

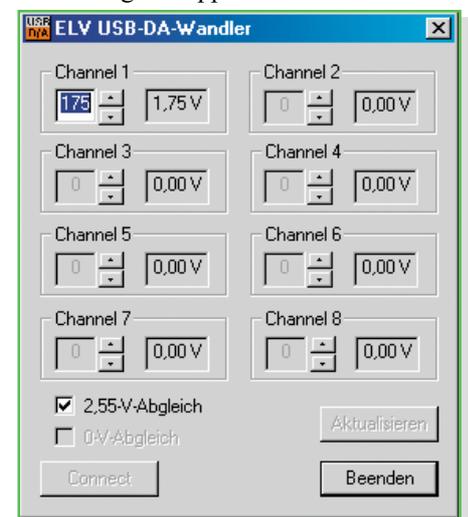


Bild 9: Der Abgleich-Modus des UDA 7