

# Serieller Datenlogger



## für USB-Speicher-Sticks

Viele Multimeter, GPS-Empfänger und andere Geräte besitzen auch heute noch EIA232-Schnittstellen, um empfangene oder gemessene Daten ausgeben zu können. Leider ist, trotz des einfachen Handlings dieser Schnittstelle, für den Empfang und das Speichern von übertragenen Daten der Aufwand eines angeschlossenen PCs notwendig. Dies macht gerade den mobilen Einsatz solcher Geräte kompliziert und unflexibel. Hier soll nun der serielle Datenlogger SDU 1 in die Bresche springen, der alle Daten auf einer EIA232-Schnittstelle mitloggt und direkt auf einen USB-Stick speichert.

### Technische Daten: SDU 1

Schnittstelle:	serielle EIA232 (RS232)
Übertragungsgeschwindigkeit: (nicht kontinuierlich)	bis 115.200 Bit/s
Unterstützte Datenspeicher:	handelsübliche USB-Sticks (bis 4 GB)
Unterstützte Dateisysteme:	FAT und FAT32
Anzeigeelement:	rote LED für Aktivitätsanzeige
Bedienelemente:	8fach DIP-Schalter und Taster
Fortlaufende Dateinummerierung:	von 1 bis 100 Mio. im EEPROM gespeichert
Einstellungen:	über DIP-Schalter oder Konfigurationsdateien
Spannungsversorgung:	5 Vdc (USB-powered) über USB-Mini-B-Buchse
Stromaufnahme:	max. 250 mA (je nach USB-Stick)
Abmessungen (B x H x T):	51 x 26 x 92 mm
Im Lieferumfang:	Adapterstecker zum „Mitloggen“

### Daten speichern ohne Ende

Geräte mit seriellen Schnittstellen für die Datenausgabe sind nach wie vor weit verbreitet, die Schnittstelle ist einfach beherrschbar und für Software-Entwickler ebenso einfach programmierbar. Zudem gibt es für eigene Applikationen kaum einen Mikrocontroller, der keine serielle Schnittstelle aufweist.

Solche Geräte, wie Multimeter oder GPS-Empfänger, werden vorwiegend für das Protokollieren und Aufzeichnen von Messdaten eingesetzt, sind allerdings, falls sie nicht über einen eigenen Datenspeicher verfügen, an das Vorhandensein eines PCs gebunden.

Es gibt zwar Datenlogger für die serielle Schnittstelle, deren Speicher eignen sich aufgrund der meist eher geringen und nicht erweiterbaren Kapazität nur selten für die Langzeiterfassung. Angesichts der geradezu riesigen Speicherkapazitäten mobiler Datenspeicher wie USB-Sticks, SD-/MMC-Cards oder Compact-Flash- und anderer Speicherkarten ist eine sol-

che Technik mit fest verbaubem Flash-Speicher nicht mehr up to date. Zudem muss man beim traditionellen Datenlogger stets den gesamten Datenlogger zum Auslesen transportieren, die Messwerterfassung ruht zwischenzeitlich.

Da liegt es natürlich nahe, die erwähnten modernen Speichermedien für die autarke Datenspeicherung einzusetzen. Sie haben auch den Vorteil, mobil zu sein – man muss also nicht mehr den Messaufbau „auseinanderreißen“, um die Daten auslesen zu können. Lediglich der Daten-Speicher, in unserem Falle der USB-Stick, wird ausgetauscht und die Messung kann fortgesetzt werden. Die bisher gesammelten Daten können irgendwo auf einem entfernten PC wie üblich ausgelesen und ausgewertet werden.

Der hier vorgestellte serielle Datenlogger ist genau auf diese Aufgabe zugeschnitten! Er wird einfach mit einem handelsüblichen USB-Stick (bis 4 GB) bestückt und kann so nahezu unbegrenzt Daten über seine EIA232-Schnittstelle aufzeichnen. Das geht in zwei Varianten:

### Loggen als Endgerät

Hier erfolgt der Anschluss des Datenloggers direkt an ein Multimeter, einen GPS-Empfänger oder ein anderes Gerät mit EIA232-Schnittstelle. Der Datenlogger empfängt die seriellen Daten und speichert sie auf dem USB-Stick. Diese Variante ist in Abbildung 1 in einer Beispiel-Konfiguration zu sehen.

### Loggen als Mithörer

Die zweite Betriebsvariante ist das Mitloggen eines seriellen Datenstroms über einen mit dem Datenlogger mitgelieferten Adapterstecker. Damit kann man die gesamte Kommunikation auf einer Leitung der verbundenen Gerätekonfiguration mitloggen. Diese Variante ist in Abbildung 2 dargestellt.

## Vielseitig

Der Datenlogger ist für eine möglichst universelle und praktische Verwendbarkeit ausgelegt, um den unterschiedlichsten Anwenderwünschen gerecht zu werden.

Im **Aufnahme-Modus (Record)** erfolgt die Speicherung der empfangenen Daten auf dem USB-Stick im Rohformat (Byte für Byte), ganz wie sie an der Schnittstelle auftreten.

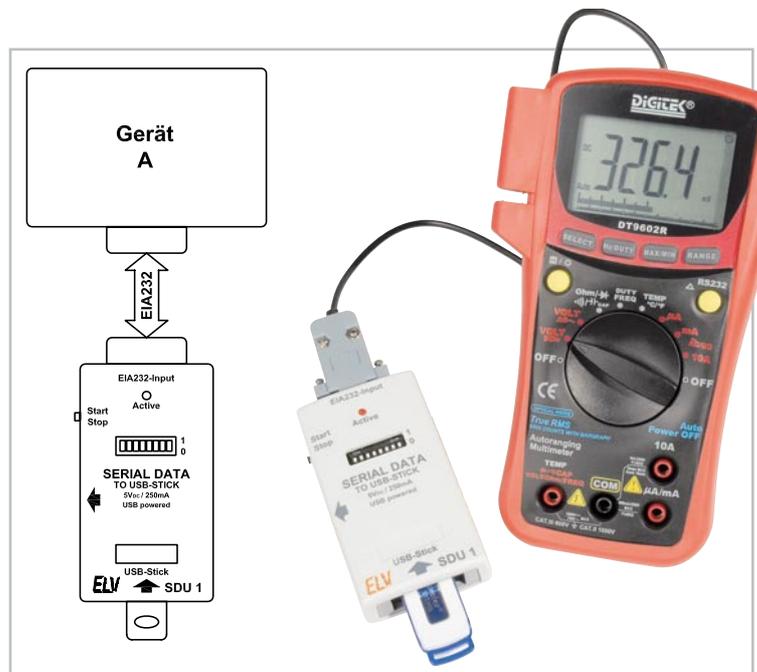


Bild 1: Der SDU 1 speichert die Messdaten eines Multimeters auf den USB-Stick.

Die eingestellte Konfiguration der Schnittstelle wird in einer Log-Datei gespeichert.

Auch der umgekehrte Weg ist möglich: Im **Wiedergabe-Modus (Play)** können die gespeicherten Daten über die serielle Schnittstelle auch wieder ausgegeben werden – ideal, um etwa Gerätekonfigurationen und Messdaten ohne eine direkte Kabelverbindung auf andere Geräte zu übertragen.

Die Schnittstelle ist jederzeit variabel über acht DIP-Schalter konfigurierbar (Übertragungsgeschwindigkeit, Stopp-Bit, Parität, Aufnahme-/Wiedergabe-Modus).

Zusätzlich sind drei unterschiedliche Konfigurationseinstellungen auch über Konfigurationsdateien möglich. Die dafür nötigen config-Dateien werden mit einem kleinen Programmtool (als kostenfreier Download erhältlich) auf einem Windows-PC erstellt und auf dem USB-Stick gespeichert. Die jeweils gewünschte config-Datei ist einfach über die DIP-Schalter des Datenloggers wählbar. Über diese config-Datei sind zusätzliche Aufzeichnungs-/Wiedergabe-Funktionen möglich (z. B. wählbarer Dateiname für die Wiedergabe-Datei, Anzahl der Datenbits wählbar, weitere Baudraten).

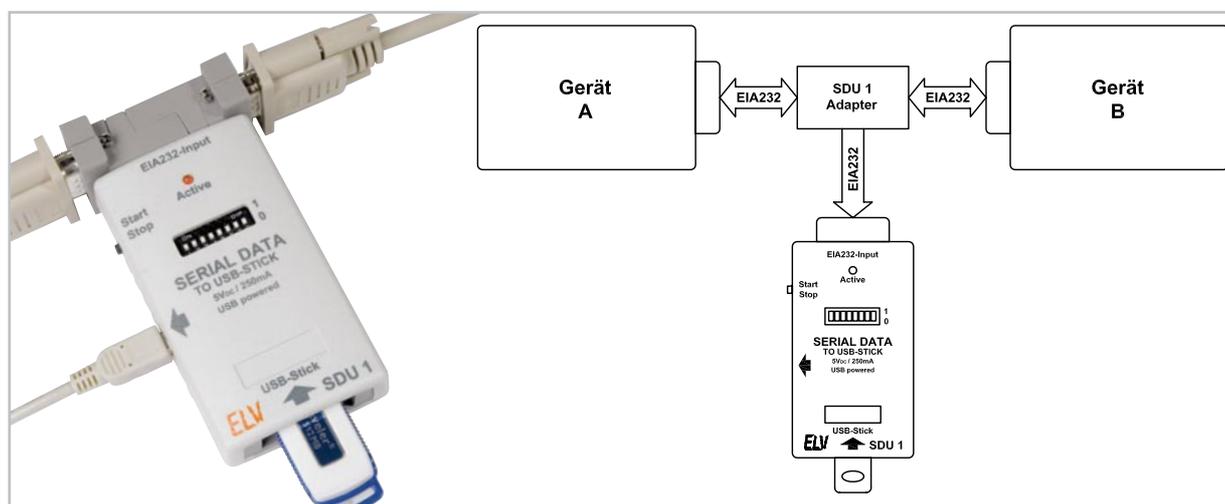


Bild 2: Mit Hilfe des Adaptersteckers speichert der SDU 1 den Datenfluss auf einer durchgeführten Leitung.

Ohne das Kapitel „Bedienung“ vorwegzunehmen, ein Satz zur einfachen Bedienung des Gerätes: Nach der Konfiguration ist lediglich das Starten/Stoppen der Datenaufnahme/-wiedergabe über den Taster „Start/Stop“ notwendig, die LED „Active“ zeigt die Aktivitäten des Datenloggers an. Die Stromversorgung des Gerätes erfolgt über eine USB-Mini-Buchse (+5 V) entweder über ein Netzteil mit USB-Mini-B-Stecker (ELV-Best.-Nr.: 84-724-96), einen Kfz-USB-Spannungswandler, wie man ihn z. B. für das Laden von mobilen Navigationsgeräten, Mediaplayern oder Handys im Auto verwendet, über einen aktiven USB-Hub oder über einen PC. Dabei kommt ein normales USB-Kabel mit Mini-USB-Stecker auf einer Seite zum Einsatz. Über einen entsprechend leistungsfähigen Akku ist sogar mobiler Betrieb möglich, entsprechende Akku-Packs gibt es z. B. für das Nachladen von Handys oder zur Versorgung von USB-Geräten.

Als Datenspeicher kommen handelsübliche USB-Sticks mit USB-1.1/2.0-Schnittstelle zum Einsatz. Kurze USB-Speicher-Sticks (Gehäuselänge <35 mm) passen vollständig ins SDU-1-Gehäuse, ohne hervorzustehen. Längere USB-Sticks sind durch die Gehäuseeinführung gegen versehentliches Abbrechen bzw. Herausziehen geschützt.

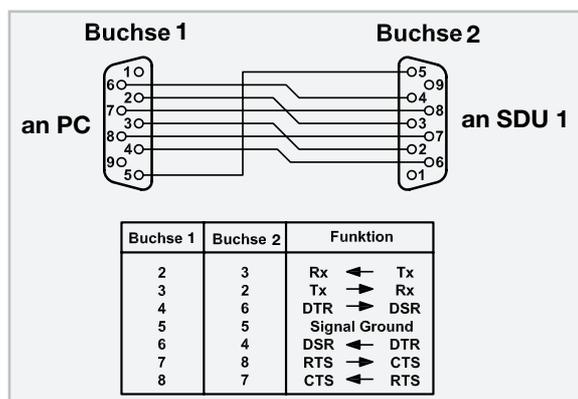
## Anwendungen

Wie bereits angedeutet, sind die Einsatzmöglichkeiten des SDU 1 sehr vielfältig. Wir wollen hier einige davon kurz betrachten.

Da wäre zunächst der Anschluss eines GPS-Empfängers (GPS-Maus) und das Aufzeichnen von Koordinaten, Geschwindigkeit, Richtung usw. z. B. im Auto, um eine gefahrene Strecke später nachvollziehen zu können. Derartige GPS-Daten können von diversen PC-Programmen weiterverwendet und z. B. in einer Google™-Maps-Karte visualisiert werden.

Ein weiteres Anwendungsgebiet ist die Speicherung von Messdaten. Sei es, um Langzeitmessungen aufzuzeichnen oder um Messdaten für Dokumentation, Kontrolle oder Qualitätssicherung zu archivieren.

Auch die bereits erwähnte Möglichkeit, den Datenverkehr zwischen 2 Geräten aufzeichnen zu können, bietet interessante Möglichkeiten. So können z. B. Messdaten mitgelesen oder Steuerbefehle aufgezeichnet werden, um z. B. Pro-



**Bild 3:** Gekreuzte Beschaltung eines Nullmodem-Kabels, wie es für den direkten Anschluss des SDU 1 an einen PC nötig ist, wobei nur die Leitungen 2, 3 und 5 gebraucht werden.

grammabläufe zu überprüfen. Auf diese Weise kann auch ein Kommunikationsprotokoll analysiert werden, über das keine genaue Dokumentation vorliegt.

Ein ähnliches Gebiet ist die mögliche Aktivierung/Bedienung von Geräten, die über EIA232 gesteuert werden, ohne einen PC anschließen zu müssen. Das kann entweder durch Wiedergabe von Befehlen geschehen, die auch zuvor mit dem SDU 1 aufgezeichnet werden können oder die der Anwender am PC in eine Datei speichert, die vom Datenlogger anschließend beliebig oft wiedergegeben werden kann.

Schließlich ist der SDU 1 auch als geradezu gigantischer Datenspeicher für eigene Mikrocontroller-Anwendungen nutzbar. Fast jeder Mikrocontroller verfügt über mindestens eine serielle Schnittstelle, die mit sehr geringem Aufwand (siehe auch das Schaltbild des SDU 1) in eine normgerechte EIA232-Schnittstelle umgewandelt und so mit dem SDU 1 verbunden werden kann. Gegenüber den sonst üblichen schmalen Speicherkapazitäten bei µC-Schaltungen eröffnen sich über die bis zu mehreren Gigabyte großen USB-Speicher ganz neue und vor allem einfach zu handhabende Einsatzmöglichkeiten für eigene Mikrocontroller-Applikationen!

## Installation und Bedienung

Die Inbetriebnahme und Bedienung des Datenloggers ist denkbar einfach.

Nach dem Anschließen der Spannungsversorgung am SDU 1 über die seitliche USB-Mini-Buchse blinkt die LED „Active“ beim Systemstart einmal kurz auf.

Wird nun der Speicher-Stick in den USB-Port des SDU 1 gesteckt, blinkt die LED nochmals auf, wenn der Speicher-Stick ordnungsgemäß erkannt wurde. Dieser sollte im FAT- oder FAT32-Dateisystem formatiert sein, was sich am PC schnell über „Laufwerksname->Eigenschaften“ überprüfen lässt.

Das Gerät, dessen Daten geloggt oder an das Daten gesendet werden sollen, wird entweder direkt (siehe Abbildung 1) oder über den mitgelieferten Adapter (siehe Abbildung 2) am SDU 1 angeschlossen. Soll der SDU 1 über die serielle Schnittstelle direkt mit einem PC verbunden werden, so ist dafür ein gekreuztes Kabel, ein sogenanntes Nullmodem-Kabel, nötig. Die Beschaltung eines solchen Kabels ist aus Abbildung 3 ersichtlich.

Über die acht DIP-Schalter ist jetzt die Schnittstelle entsprechend den Anforderungen des angeschlossenen Gerätes zu konfigurieren. Alternativ kann man zur Einstellung eine von drei möglichen am PC erstellten und auf dem USB-Stick gespeicherten config-Dateien nutzen.

Zum Starten der Datenaufzeichnung bzw. der Datenwiedergabe ist nun nur noch die „Start/Stop“-Taste zu drücken. Jetzt blinkt die LED mehrfach auf, was bedeutet, dass das Logfile erstellt bzw. erweitert wird. Sobald die LED konstant leuchtet, ist der SDU 1 bereit zur Datenaufzeichnung bzw. -wiedergabe.

## Aufnahme-Modus (Record)

Die empfangenen Daten werden auf dem USB-Stick in einer neu angelegten Rohdaten-Datei gespeichert. Als Dateiname wird eine fortlaufende Nummer gewählt, die mit

„00000001.DAT“ beginnend mit jeder neuen Aufnahme um eins hochgezählt wird.

Zurücksetzen lässt sich dieser im EEPROM gespeicherte Dateizähler, indem alle acht DIP-Schalter in die obere Position auf „On“ gestellt werden und anschließend die „Start/Stop“-Taste kurz gedrückt wird. Anzeigt wird das Rücksetzen durch ein dreimaliges Aufblinken der LED.

Existiert eine Datei bereits, so wird diese nicht überschrieben, sondern um die neuen Daten erweitert, indem diese am Ende angehängt werden.

**Achtung:** Jede Aufzeichnung muss durch Drücken der „Start/Stop“-Taste beendet werden, da sonst die gerade geschriebene Datei nicht abgeschlossen wird und die gespeicherten Daten verloren gehen. Nach dem Drücken des Tasters ist mit dem Rausziehen des USB-Sticks noch so lange zu warten, bis die LED erloschen ist.

Der USB-Stick darf auf keinen Fall aus dem SDU 1 herausgezogen werden, solange die LED leuchtet, da in diesem Fall ein Datenverlust vorprogrammiert ist. Auch beim Abtrennen der Stromversorgung würden alle Daten der gerade geschriebenen Datei verloren gehen. (Im Fall des Datenverlustes kann eventuell ein Teil der Daten mit dem Windows-Kommandozeilenprogramm „CHKDSK“ wiederhergestellt werden.)

### Wiedergabe-Modus (Play)

Nach dem Start werden die Daten aus der zuletzt aufgenommenen Datei über die serielle Schnittstelle ausgegeben – allerdings nur, wenn diese Datei auch existiert und Daten enthält.

Möchte man statt der letzten Datendatei die vorletzte oder eine noch weiter zurückliegende Datei ausgeben, so ist der Taster zum Start so lange gedrückt zu halten, bis die LED zuerst wieder erlischt und dann periodisch kurz aufflackert. Je nachdem wie häufig man die LED aufflackern lässt, wird mit jedem Blinken eine weiter zurückliegende Datei gewählt (1x Blinken = vorletzte Datei, 2x Blinken = vorvorletzte Datei usw.). Nach dem Loslassen des Tasters wird diese ausgegeben. Nachdem eine Datei vollständig ausgegeben wurde, kann dieselbe Datei noch mal wiedergegeben werden, indem die Taste wiederum gedrückt gehalten wird, bis die LED von selbst erlischt. Möchte man noch weiter zurück, so wartet man vor dem Loslassen der Taste wie zuvor das Blinken ab. Sobald die Taste zum Start aber nur kurz gedrückt wird, springt der SDU 1 wieder zur neusten Datei zurück und gibt deren Inhalt wieder.

Mit Hilfe des Konfigurationstools kann zuvor am PC auch eine bestimmte Datendatei anhand ihres Dateinamens für die Wiedergabe ausgewählt werden.

Nachdem ein Datenfile einmal komplett gesendet ist, wird die Wiedergabe automatisch beendet.

Sowohl die Datenaufzeichnung als auch die Datenwiedergabe können jederzeit durch nochmaliges Drücken der „Start/Stop“-Taste beendet werden.

Die auf dem USB-Stick gespeicherten Daten können anschließend direkt am PC ausgewertet werden. Dazu ist (nach der Beendigung der Datenaufnahme und dem Erlöschen der LED) der USB-Stick aus dem SDU 1 zu entfernen und in den PC zu stecken. Da die gespeicherten Daten entsprechend dem zu vor angeschlossenen Gerät unterschiedlichste Formate haben

**Tabelle 1: Über die DIP-Schalter einstellbare Übertragungsgeschwindigkeiten**

DIP-Schalter								Mögliche Übertragungs-Bitraten
1	2	3	4	5	6	7	8	
0	0	0	0	x	x	x	x	300 bit/s
0	0	0	1	x	x	x	x	600 bit/s
0	0	1	0	x	x	x	x	1200 bit/s
0	0	1	1	x	x	x	x	2400 bit/s
0	1	0	0	x	x	x	x	4800 bit/s
0	1	0	1	x	x	x	x	9600 bit/s
0	1	1	0	x	x	x	x	14.400 bit/s
0	1	1	1	x	x	x	x	19.200 bit/s
1	0	0	0	x	x	x	x	28.800 bit/s
1	0	0	1	x	x	x	x	38.400 bit/s
1	0	1	0	x	x	x	x	57.600 bit/s
1	0	1	1	x	x	x	x	76.800 bit/s
1	1	0	0	x	x	x	x	115.200 bit/s

**Tabelle 2: Über die Konfigurationsschalter einstellbare Stopp-Bits**

DIP-Schalter								Anzahl Stopp-Bits
1	2	3	4	5	6	7	8	
x	x	x	x	x	x	x	0	1 Stopp-Bit
x	x	x	x	x	x	x	1	2 Stopp-Bits

**Tabelle 3: Über die Konfigurationsschalter einstellbare Parität**

DIP-Schalter								Parität
1	2	3	4	5	6	7	8	
x	x	x	x	x	0	0	x	keine (none)
x	x	x	x	x	0	1	x	gerade (even)
x	x	x	x	x	1	0	x	ungerade (odd)

**Tabelle 4: Über die Konfigurationsschalter einstellbare Datenrichtung**

DIP-Schalter								Datenrichtung (Aufnahme/Wiedergabe)
1	2	3	4	5	6	7	8	
x	x	x	x	0	x	x	x	in Datei speichern (record)
x	x	x	x	1	x	x	x	aus Datei senden (play)

können, bleibt dem Anwender die Auswertung über eine geeignete Software überlassen.

Die Daten eines GPS-Empfängers können beispielsweise direkt mit einem Texteditor angeschaut werden. Dazu sollte man eventuell die Endung der Datei von .dat auf .txt umbenennen.

Dasselbe gilt für die Daten vieler Messgeräte, auch hier erfolgt oft das Ablegen als (ASCII-)Textdatei. Diese können häufig jedoch auch mit Excel importiert und weiterverarbeitet werden.

Daten, die nicht im ASCII-Format codiert sind, kann man mit einem Hex-Editor wie beispielsweise WinHEX oder HTerm auswerten. Alle empfangenen Datenbytes werden unformatiert und aufeinander folgend in die jeweils neuste Datei gespeichert.

In den Tabellen 1 bis 5 sind alle direkt am SDU 1 vorzunehmenden Konfigurationseinstellungen aufgelistet. Um diese Einstellungen jederzeit parat zu haben, sind diese auch auf der Geräterückseite aufgedruckt.

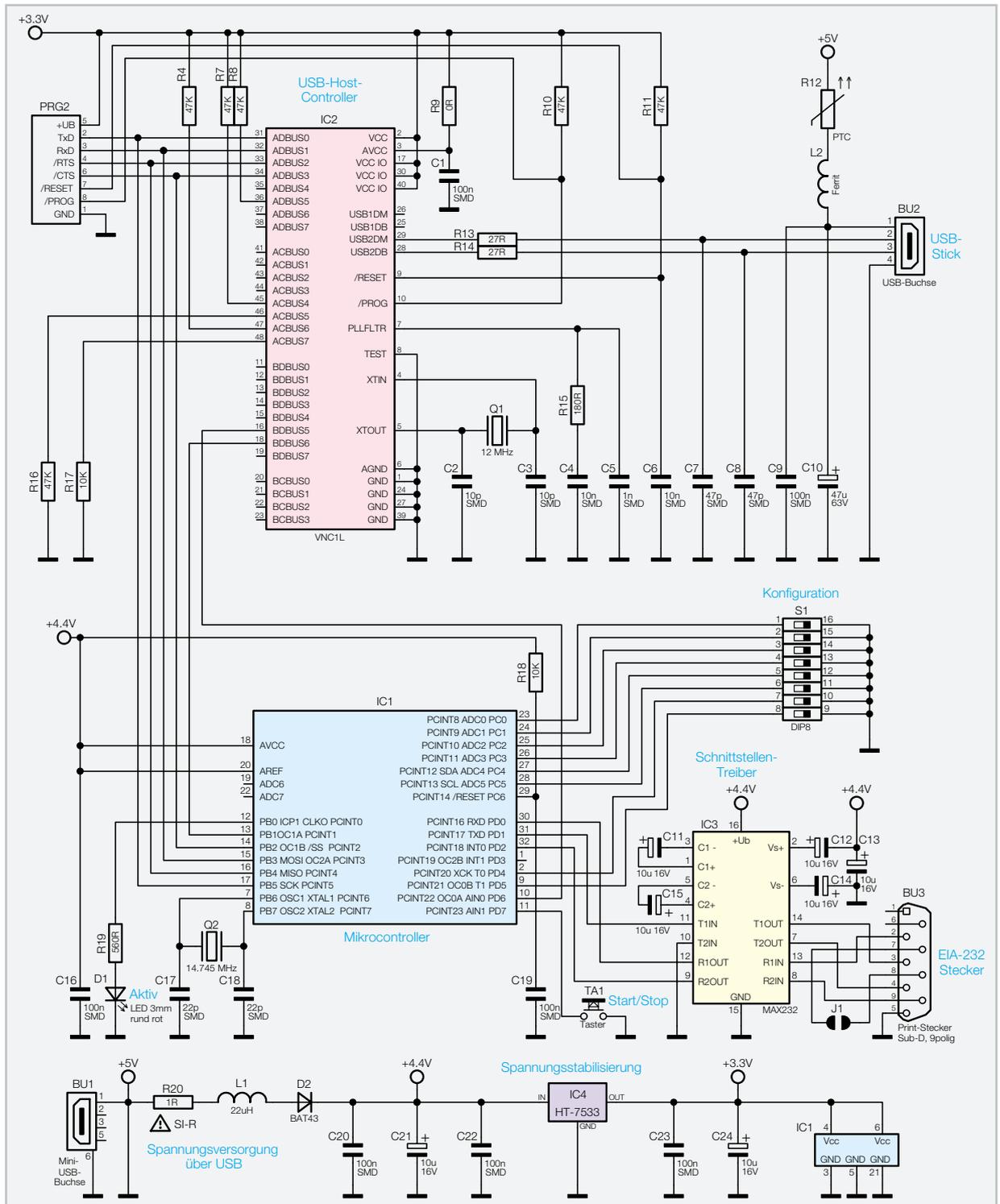


Bild 4: Das Schaltbild des SDU 1

Neben diesen Einstellmöglichkeiten gibt es noch einige zusätzliche Optionen, die jedoch nur über das Konfigurationsprogramm wählbar sind, das über die Internet-Angebotsseite des SDU 1 kostenfrei zum Download bereitsteht.

Tabelle 5: DIP-Schalterstellung für Konfiguration über config-Dateien

DIP-Schalter								Konfiguration aus Datei lesen
1	2	3	4	5	6	7	8	
x	x	x	x	0	1	1	0	Config1.ini
x	x	x	x	0	1	1	1	Config2.ini
x	x	x	x	1	1	1	1	Config3.ini

## Schaltung

Das Schaltbild des SDU 1 ist in Abbildung 4 dargestellt. Als zentrales Element dient der ATmega88-Mikrocontroller IC 1, dem IC 2, ein VNC1L-USB-Host-Controller von Vinculum, für den USB-Speicherzugriff zur Seite steht. Verbunden sind die beiden Bausteine über eine schnelle SPI-Schnittstelle, die mit 3,7 MHz getaktet wird. Theoretisch könnte der Datenlogger über diese schnelle Verbindung auch höhere Bitraten problemlos und kontinuierlich verarbeiten, jedoch benötigt der VNC1L für das Speichern auf dem USB-Stick im FAT/FAT32-Dateisystem Zugriffszeiten von zeitweise mehr als

**Bild 5:** Das Schaltbild des SDU 1-Adaptersteckers. Geloggt werden die Daten auf Leitung 2.

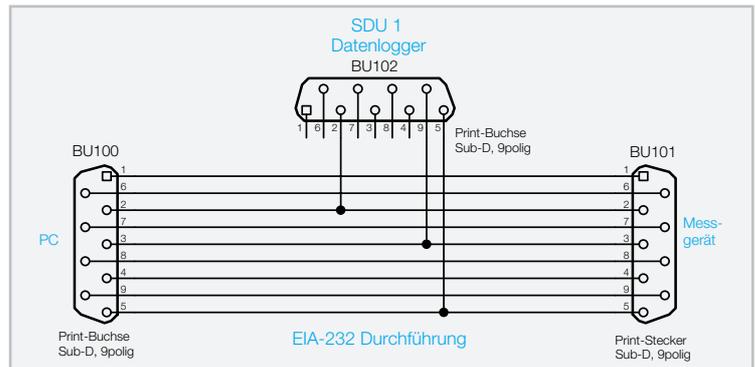
500 ms. Während dieses Zeitraums muss der ATmega88 alle eintreffenden Daten zwischenspeichern, wofür ihm nur ein begrenzter Speicherbereich zur Verfügung steht.

Sehr vorteilhaft ist die hohe Integration vieler Funktionselemente im VNC1L. USB-Speicher-Sticks lassen sich direkt an IC 2 anschließen und über einen einfachen Befehlssatz beschreiben und auslesen. Extern werden lediglich der 12-MHz-Quarz Q 1, die zur Konfiguration des USB-Host-Controllers dienenden Pull-up/down-Widerstände (R 4, R 7, R 8, R 9, R 10, R 11, R 16, R 17) und eine Reihe von Kondensatoren zur Spannungsglättung (C 24, C 10) und zur Störungsunterdrückung (C 1, C 6, C 7, C 8, C 9, C 20, C 22, C 23) benötigt. Das PTC-Sicherheitselement R 12 begrenzt im Fehlerfall den an der USB-Buchse BU 2 zur Verfügung stehenden Strom.

Der VNC1L benötigt eine eigene angepasste Firmware, die über die spezielle Programmierschnittstelle PRG 2 bereits werkseitig implementiert wurde.

Die Versorgung des USB-Host-Controllers IC 2 erfolgt durch die mit IC 4 stabilisierte 3,3-V-Gleichspannung, während IC 1 seine Betriebsspannung direkt aus der 5-V-Eingangsspannung erhält. Durch die Diode D 2 wird lediglich der Spannungspegel auf ca. 4,4 V abgesenkt, um eine sichere SPI-Verbindung zwischen IC 1 und IC 2 zu gewährleisten. Am Eingang von IC 2 ist zwar auch ein Spannungspegel von 5 V zulässig, jedoch liegt umgekehrt eine 3,3-V-Eingangsspannung bereits im Grenzbereich eines zulässigen High-Pegels des ATmega88. Theoretisch könnte IC 1 zwar ebenfalls mit 3,3 V betrieben werden, allerdings nicht bei der hohen Taktrate von 14,7456 MHz.

Eine an der USB-Mini-Buchse BU 1 anliegende Eingangsspannung wird direkt zur Buchse BU 2 und damit auch an einen angeschlossenen USB-Stick weitergeleitet. Aus diesem



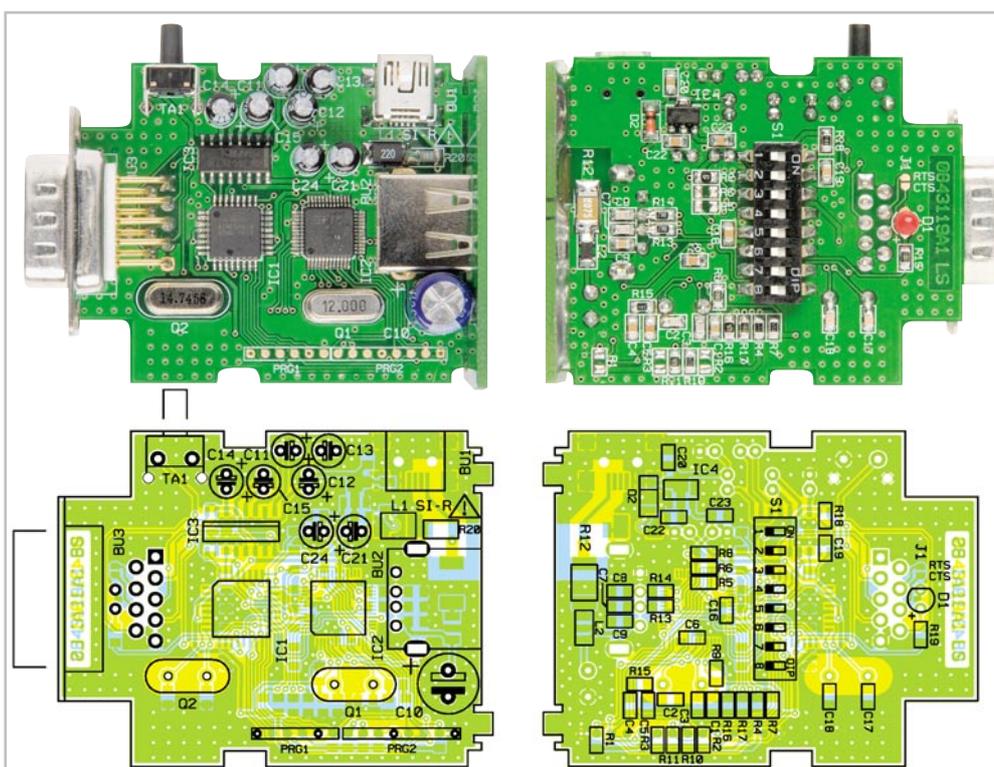
Grund muss unbedingt darauf geachtet werden, dass ausschließlich stabilisierte und geglättete 5-V-Spannungsquellen verwendet werden. Die Buchse BU 1 hat keine weitere USB-Funktionalität und dient nur zur Versorgung des SDU 1 über einen angeschlossenen PC oder über andere USB-Spannungsquellen.

Als Bedienelemente dienen der an IC 1 angeschlossene 8fach-DIP-Schalter S 1, der Taster TA 1 und die rote LED D 1.

Die EIA232-Schnittstelle des SDU 1 ist über den MAX232-Schnittstellentreiber IC 3 am ATmega88 angeschlossen. Für die Erzeugung der  $\pm 9$ -V-Spannung auf der EIA232-Schnittstelle benötigt dieser Baustein als externe Beschaltung die fünf Elkos C 11 bis C 15.

Die im Schaltbild als J 1 bezeichneten Löt pads können bei Bedarf miteinander verbunden werden, wenn ein angeschlossenes Gerät einen Hardware-Handshake (RTS->CTS) benötigt. Signalisiert solch ein Gerät auf der RTS-Leitung (Request to send), dass es zum Senden bereit ist, so erhält es aufgrund der J-1-Verbindung gleichzeitig auf der CTS-Leitung (Clear to send) ein O. k. fürs Senden zurück. Im Normalfall bleibt J 1 aber offen, da kaum ein Gerät Hardware-Handshakes verwendet.

In Abbildung 5 ist das Schaltbild des zugehörigen Adapter-



Ansicht der fertig bestückten Platine des SDU 1 mit zugehörigem Bestückungsplan, links von der Bestückungsseite, rechts von der Lötseite

## Stückliste: SDU 1

**Widerstände:**

0 Ω/SMD/0805	R9
Sicherungswiderstand 1 Ω/SMD/1206	R20
27 Ω/SMD/0805	R13, R14
180 Ω/SMD/0805	R15
560 Ω/SMD/0805	R19
10 kΩ/SMD/0805	R17, R18
47 kΩ/SMD/0805	R4, R7, R8, R10, R11, R16
Polyswitch, 13,2 V, 0,75 A, SMD, 1812	R12

**Kondensatoren:**

10 pF/SMD/0805	C2, C3
22 pF/SMD/0805	C17, C18
47 pF/SMD/0805	C7, C8
1 nF/SMD/0805	C5
10 nF/SMD/0805	C4, C6
100 nF/SMD/0805	C1, C9, C16, C19, C20, C22, C23
10 µF/16 V	C11–C15, C21, C24
47 µF/63 V	C10

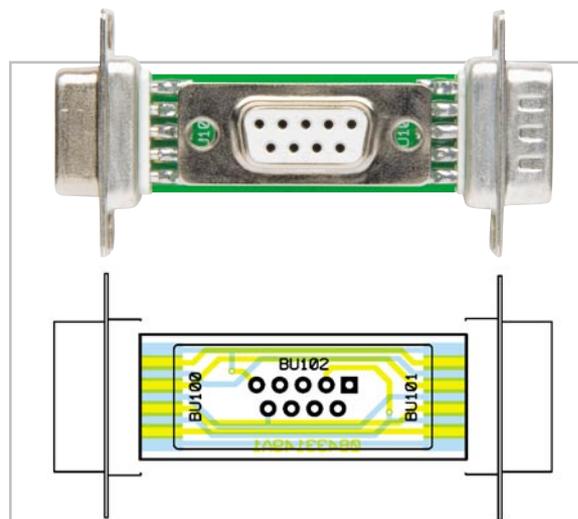
**Halbleiter:**

ELV08760/SMD/Hauptcontroller	IC1
ELV08761/SMD/USB-Host-Controller	IC2
MAX232D/SMD	IC3
HT7533/SMD	IC4
BAT43/SMD	D2
LED, 3 mm, Rot	D1

**Sonstiges:**

Quarz, 12 MHz, HC49U4	Q1
Quarz, 14,745 MHz, HC49U4	Q2
SMD-Induktivität, 22 µH	L1
Chip-Ferrit, 1206, 80 Ω bei 100 MHz	L2
USB-B-Buchse, mini, 5-polig, winkelprint, liegend, SMD	BU1
USB-A-Buchse, winkelprint, liegend	BU2
SUB-D-Stiftleiste, 9-polig, abgewinkelt	BU3
SUB-D-Buchsenleiste, 9-polig, Lötanschluss	BU100
Sub-D-Stiftleiste, 9-polig, Lötanschluss	BU101
SUB-D-Buchsenleiste, 9-polig, print, gerade	BU102
Mini-Taster, abgewinkelt, print	TA1
Mini-DIP-Schalter, 8-polig, liegend, SMD	S1
1 Element-Gehäuse, G431, komplett, Hellgrau, bearbeitet und bedruckt	
1 SUB-D-Leergehäuse, 2 x 9-polig, bearbeitet	

steckers für das Mitloggen einer Datenverbindung dargestellt. Einfach zu erkennen ist die direkte Durchleitung aller Anschlüsse von BU 100 zu BU 101. Diese Verbindung funktioniert also wie ein Stück „Verlängerungskabel“. Über die



Fertig bestückte Buchsenplatine mit zugehörigem Bestückungsplan

Buchse BU 102 kann ein angeschlossener SDU 1 den Datenverkehr auf der Datenleitung (Pin 2) mitloggen, ohne dabei die angeschlossenen Geräte zu stören.

## Nachbau

Ein Großteil der SDU-1-Elektronik ist werkseitig bereits bestückt, da es sich um empfindliche SMD-Bauteile handelt. Einige bedrahtete Bauteile sind jedoch noch von Hand zu bestücken. Dies geschieht in gewohnter Weise mit Hilfe der Stückliste, des Bestückungsdrucks und des Schaltbildes. Mit Ausnahme von LED D 1 werden alle Bauteile mit ihren Anschlüssen von oben in die dafür vorgesehenen Bohrungen gesteckt und von unten verlötet.

Bei den Elkos C 10 bis C 15, C 20, C 21, C 24 und der Leuchtdiode D 1 ist beim Einlöten unbedingt auf die richtige Polarität zu achten. Falsch gepolte Elkos können sogar platzen. Der Minuspol der Elkos ist auf einer Seite am Gehäuse gekennzeichnet. Auf der Platine ist hingegen der gegenüberliegende Pluspol markiert. Der auf der Platine durch ein Plus gekennzeichnete Anodenanschluss der LED ist am Bauteil selbst durch den längeren Anschluss zu erkennen. Die LED-Anschlüsse werden von unten so weit durch die Platine gesteckt, bis das LED-Gehäuse aufliegt, und von oben verlötet.

Bei den Quarzen ist darauf zu achten, dass in die mit Q 1 bezeichneten Bohrungen der 12-MHz-Quarz und in Q 2 der 14,745-MHz-Quarz gelötet wird.

Die EIA232-Buchse BU 3 ist, wie in Abbildung 6 zu sehen, so einzulöten, dass deren Metallgehäuse auf dem vorderen Platinenrand aufliegt und die Lötstifte rechtwinklig durch die Bohrungen ragen. Der Taster TA 1 wird ebenfalls von oben

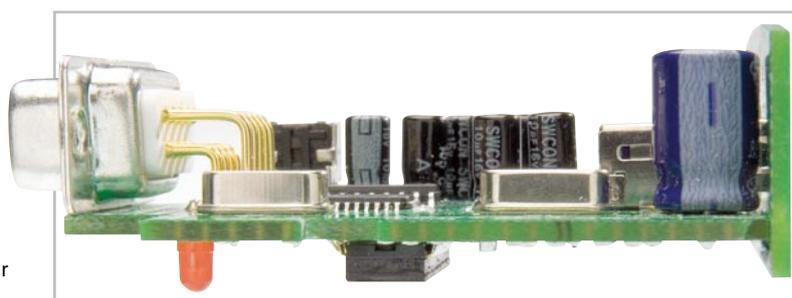
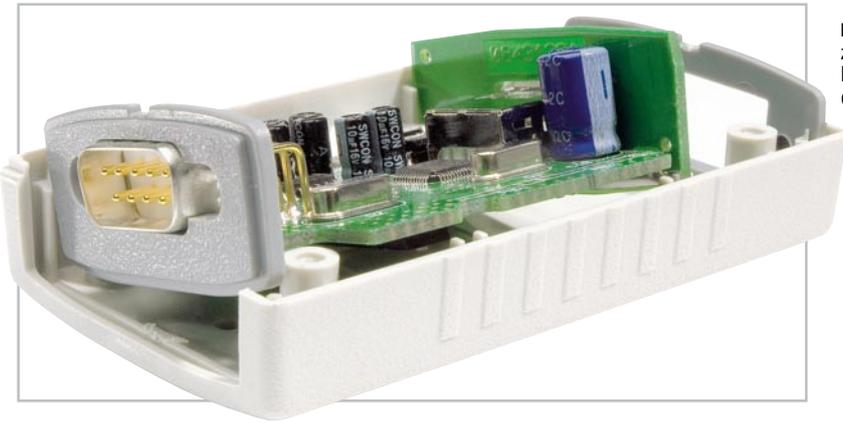


Bild 6: Seitliche Ansicht der bestückten SDU-1-Platine



**Bild 7:** Für den Einbau der Platine muss diese zusammen mit der Frontplatte seitlich in die Gehäuseschale hineingedreht werden. Vorher ist der Tasterkopf in seine Bohrung zu stecken.

in die Platine gesteckt und von unten an allen vier Lötstiften verlötet.

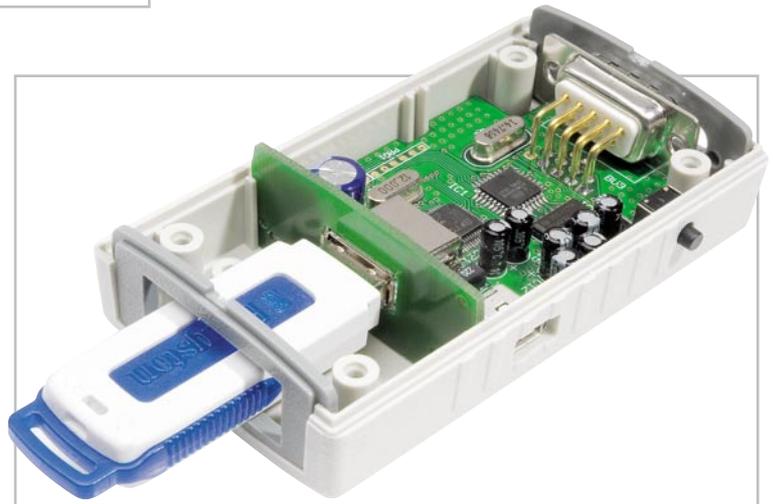
Bevor die rechts in Bild 6 abgebildete kleine Platine rechtwinklig an die Hauptplatine gesteckt wird, muss die große USB-Buchse BU 2 in die vorgesehenen Bohrungen gesetzt (genau ausrichten und ohne Hilfsmittel fest hineindrücken) und verlötet werden.

Anschließend wird die kleine Abschlussplatine vorsichtig über den Metallkragen der Buchse BU 2 gestülpt und so auf die Hauptplatine gesteckt, dass die kleinen Platinennasen in die länglichen Schlitzte ragen. Die Platinen sind nun rechtwinklig auszurichten und von unten miteinander zu verlöten.

Nun erfolgt der Zusammenbau von Platine und Gehäuse. Dafür ist zuerst die oval ausgefräste graue Frontabdeckung, wie in Abbildung 7 zu sehen, auf die EIA232-Buchse BU 3 zu setzen. Als Nächstes wird die Platine seitlich gekippt und mit dem Tasterkopf und der Mini-USB-Buchse BU 1 durch die (im Bild nicht zu sehenden) passenden Bohrungen im Gehäusedeckel gesteckt. Die graue Frontabdeckung und die senkrecht aufgelötete kleine Platine setzt man dabei in ihre Führungsrillen ein.

Nun wird die Hauptplatine samt Frontplatte und senkrechter Platine seitlich in die Führungen der Gehäuseschale hineingedreht, bis Platine und Frontplatte gerade im Gehäuse sitzen.

Abbildung 8 zeigt die richtig eingesetzte Platine, die in den Führungsrillen sitzende senkrechte kleine Platine und den durch seine Bohrung ragenden Tasterkopf. Zudem ist auf die-



**Bild 8:** Richtig eingesetzte Platine mit eingestecktem USB-Stick

sem Bild zu erkennen, wie ein USB-Stick im Innern des Gehäuses in der USB-Buchse steckt. Für übergroße USB-Sticks, die nicht mehr durch die rechteckige Fräsung der rückseitigen Abdeckung passen, kann man diese Rückplatte auch komplett weglassen.

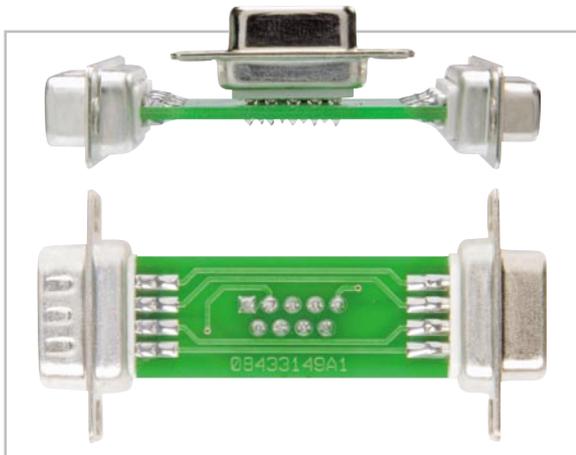
Abgeschlossen wird der Zusammenbau des Datenloggers mit dem Aufsetzen und Festschrauben der Gehäuseunterschale (der SDU 1 liegt in Abbildung 7 und 8 auf dem Kopf).

Als Nächstes erfolgt der Zusammenbau des SDU-1-Adaptersteckers. Dafür ist, wie in Abbildung 9 zu sehen, die Buchse BU 100 mit den Lötkelchen von links bis zum Anschlag auf die kleine Adapterplatine zu schieben und so anzulöten, dass die Lötkelche genau mittig auf den jeweiligen Lötspads liegen. Dabei befinden sich oben 5 und unten 4 Lötkelche.

Von rechts wird in gleicher Weise der Stecker BU 101 aufgesetzt und angelötet. Von oben wird die Buchse mit den langen dünnen Lötstiften BU 102 durch die Platine gesteckt und von unten verlötet.

Abschließend werden die beiden grauen Gehäuseschalen des Adaptersteckers von oben und unten auf die bestückte Platine gesteckt und noch nicht gleich zusammengedrückt, da zuvor noch die vier Halteschrauben links und rechts mit in die Gehäuseschalen gelegt werden. Dabei zeigen die Gewinde der Schrauben jeweils nach außen.

Wenn man möchte, kann man diese Schrauben auch weglassen, da mit diesen lediglich die angeschlossenen Stecker extra gesichert werden können. Die Gehäuseschalen werden abschließend einfach zusammengedrückt, wodurch die Halteungen einrasten. **ELV**



**Bild 9:** Seitliche Ansicht des fertig zusammengelöteten Adaptersteckers des SDU-1-Adapters