



Ethernet-UART-Gateway EUG 100

Das Gateway ermöglicht es, Geräte oder Mikrocontroller mit UART-Schnittstelle auf einfachste Weise mit einem Netzwerk zu verbinden. Das kompakte Modul setzt das RS232-Protokoll in das TCP/IP-Protokoll (und umgekehrt) um. Steuerung, Statusabfragen oder das Auslesen von Messwerten können mit Hilfe des Gateways sowohl über das lokale Netzwerk als auch über das Internet erfolgen.

Einfach umsetzen

Die Vernetzung von Geräten (z. B. Messgeräte, Wetterstationen, Ladegeräte, Datenlogger) spielt seit einiger Zeit eine immer größer werdende Rolle in der Technik, im Zuge stetig zunehmender Vernetzung auch im Privatbereich. Durch steigende Stückzahlen der Hardware (Mikrocontroller etc.) und die dadurch fallenden Preise können heute selbst kleinere Applikationen „ins Netz gebracht werden“. So sind die (Fern-) Steuerung, die Datenerfassung oder Statusabfragen auch von fernen Standorten aus kein Problem mehr.

Das direkte Ansprechen von Mikrocontrollern über das Netzwerk oder das Internet verlangt jedoch tiefere Kenntnisse in der Netzwerkprogrammierung, dies erfordert in der Regel Spezialkenntnisse und eine gewisse Erfahrung.

Auf der anderen Seite ist eine asynchrone, serielle Schnittstelle (UART) heutzutage jedoch bei (fast) allen Mikrocontrollern integriert und verhältnismäßig einfach zu programmieren. Die Implementierung einer solchen Schnittstelle in eigene Applikationen gehört zum Standardrepertoire eines jeden Mikrocontroller-Programmierers.

Um auf recht einfache Weise auf solche Geräte über das Netzwerk oder das Internet zugreifen zu können, ist ein „Datenumsetzer“ nötig – ein Ethernet-UART-Gateway, wie wir es an dieser Stelle vorstellen wollen.

Das Modul setzt das RS232-Protokoll in das TCP/IP-Protokoll um. Über das Netzwerk gesendete Nachrichten werden in serielle Daten umgesetzt und über den UART-Port an den

Mikrocontroller der eigenen Applikation ausgegeben. Dasselbe geschieht mit Daten, die von diesem Mikrocontroller über den UART-Port gesendet werden. Der Gateway-Mikrocontroller setzt diese auf das TCP/IP-Protokoll um und sendet sie an das Netzwerk/Internet.

Installation

Das Ethernet-UART-Gateway wird mit dem bestehenden Heimnetzwerk gemäß Abbildung 1 verbunden.

Heutige DSL-Router verfügen über einen DHCP-Server, der dafür sorgt, dass jedes Gerät in einem Netzwerk eine eindeutige IP-Adresse und weitere Konfigurationsparameter (Hostname, Netzmaske usw.) zugewiesen bekommt. So geschieht dies auch mit dem Ethernet-UART-Gateway, eine manuelle Zuweisung entfällt damit.

Soll das Ethernet-UART-Gateway auch außerhalb des Heimnetzwerkes erreichbar sein, so sind einige Einstellungen im

Technische Daten: EUG 100

Schnittstellen:	Ethernet (RJ45), UART (Stiftleiste)
Betriebsspannung:	5–40 V, DC/AC
Stromaufnahme:	9 mA
Abmessungen (B x H x T):	48 x 20 x 60 mm

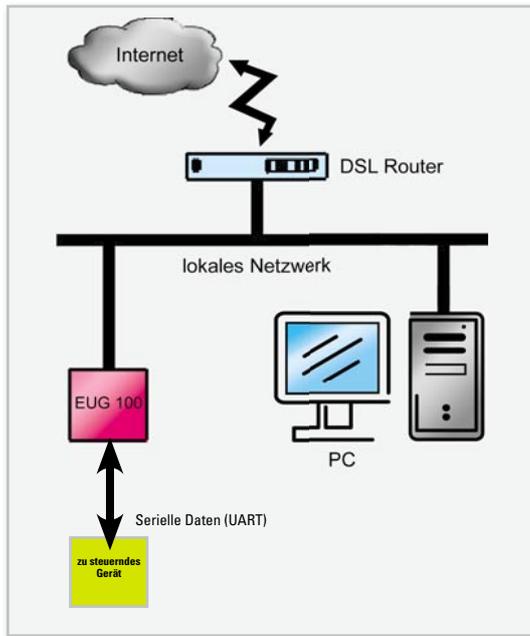


Bild 1: Einbindung des Ethernet-UART-Gateways in das bestehende lokale Netzwerk mit Anbindung an das Internet

Netzwerk vorzunehmen. Der eingesetzte DSL-Router bekommt vom DSL-Provider eine eindeutige Internet-IP-Adresse zugewiesen. Da diese Adressvergabe in der Regel dynamisch erfolgt, ist der Router nach jeder Einwahl unter einer anderen Adresse erreichbar. Abhilfe schafft hier z. B. der Service von DynDNS (<http://www.dyndns.com/>), indem er dynamische Adressen (z. B. 82.56.170.154) auf statische Adressen (z. B. dyneug100.com) umsetzt. Unter dieser statischen Adresse ist das lokale Netzwerk jetzt im Internet erreichbar, jedoch ist es nicht möglich, direkt auf die IP-Adressen innerhalb des Netzwerks zuzugreifen. Um das Ethernet-UART-Gateway dennoch zu erreichen, muss im Router eine Portweiterleitung aktiviert werden. Wie dabei vorzugehen ist, wird in der Regel in der Bedienungsanleitung des Routers beschrieben. Jetzt ist das EUG 100 außerhalb des lokalen Netzwerks durch Eingabe der statischen Adresse mit Angabe des entsprechenden Ports zu erreichen. Adresse und Portnummer sind durch einen Doppelpunkt zu trennen, z. B. <http://www.dyneug100.com:1000>.

Weiterhin ist es nötig, die Versorgungsspannung über eine Stiftleiste (ST 1) anzuschließen (Abbildung 2). Die Betriebsspannung muss zwischen 5 und 40 V liegen. Diese Betriebsspannung wird mit einem Step-down-Regler (IC 3) auf 3,3 V herabgesetzt. Weiterhin wird über die Stiftleiste ST 1 die UART-Schnittstelle zur Verfügung gestellt. Neben den Sendebzw. Empfangsleitungen (TxD und RxD) sind zwei Signalleitungen zur Steuerung der Kommunikation herausgeführt. RTS: Request to Send, signalisiert die Sendebereitschaft,

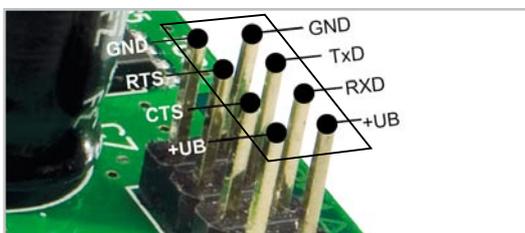


Bild 2: Pinbelegung EUG 100 (ST 1)

CTS: Clear to Send, zeigt die Sendebereitschaft des Partners an. Zu beachten ist noch, dass die Leitung vom UART-Port zum jeweils zu steuernden Gerät kürzer als 3 m ist, da sonst evtl. vorhandene äußere Störungen das Signal beeinflussen könnten. Jedoch sollte diese Leitungslänge für jeden Anwendungsfall mehr als ausreichen.

Verwendung

Um Daten austauschen zu können, ist es nötig, ein Programm zu schreiben, welches eine Verbindung mit dem Ethernet-UART-Gateway herstellt.

Die PC-Software ist die Software, welche der Anwender des Ethernet-UART-Gateways – je nach gewünschtem Anwendungsfall – selbst schreibt und deshalb auch detailliert verstehen muss.

Die Socketprogrammierung ist die Grundlage der Programmierung verteilter Anwendungen unter TCP/IP in Client-Server-Architekturen als auch bei Internetanwendungen. Ein Socket (engl. Steckdose) ist ein Verbindungsendpunkt, der vom Programm wie eine gewöhnliche Datei beschrieben und gelesen werden kann.

Dabei gibt es normalerweise ein Programm, das Anfragen von anderen Programmen entgegennimmt und sie beantwortet (ein sogenanntes Server-Socket), und Programme, die ihre Anfragen an das Server-Socket senden und mit den Antworten weiterarbeiten (die sog. Client-Sockets). Das ganze System ist auch bekannt als Client-Server-Programmierung.

Der Server (Dienstbringer) wartet darauf, dass ein Client (Kunde) mit ihm Kontakt aufnehmen möchte. Der Client ist der aktive Part und veranlasst den Beginn der Kommunikation (dieser Vorgang ist in Abbildung 3 beschrieben).

Die Socket-Netzwerkschnittstelle ist von C-Programmierern entworfen worden. Daher sind die Basisfunktionen dieser

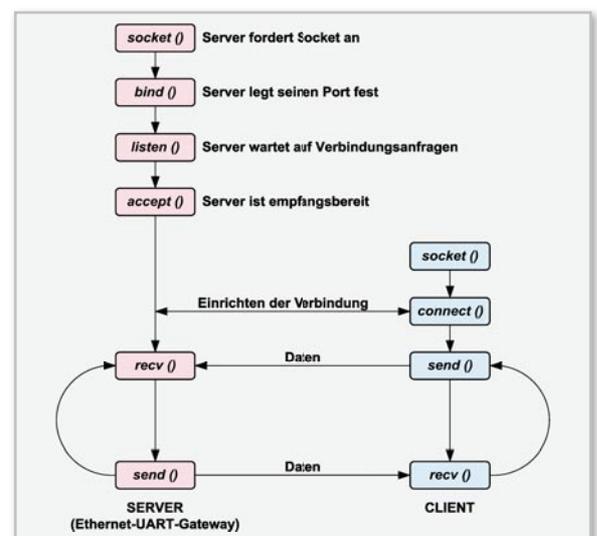


Bild 3: Kommunikation zwischen Client und Server

Programmierschnittstelle auch als C-Funktionen implementiert. Inzwischen ist die Socket-Netzwerkschnittstelle aber auch für andere Programmiersprachen verfügbar.

Im Folgenden wird ein C-Beispielprogramm beschrieben, welches Daten mit dem Ethernet-UART-Gateway austauscht.

Dieses Programm hat keinen tieferen Sinn, es dient lediglich zum Verständnis der Netzwerkprogrammierung bzw. zum Verstehen der Kommunikation mit dem Ethernet-UART-Gateway. Dieser Beispielquellcode ist auch auf der Homepage [1] zum Download bereitgestellt.

Beispielprogramm-Beschreibung

Wird der Beispielquellcode mit einem C-Compiler übersetzt, so erscheint das Ausgabefenster wie in Abbildung 4 zu sehen. Der Benutzer wird aufgefordert, die IP-Adresse des Ethernet-UART-Gateways einzugeben und kann daraufhin wählen: Zum einen können Daten über das Netzwerk zum Ethernet-UART-Gateway gesendet werden oder zum anderen können bereits empfangene Daten abgerufen bzw. angezeigt werden.

Bild 4: Client-Programm

Um die IP-Adresse des Ethernet-UART-Gateways herauszufinden, ist beispielsweise die Konfigurationsseite des DSL-Routers aufzurufen. Auf dieser Konfigurationsseite ist abzulesen, welches angeschlossene Gerät welche IP-Adresse vom DHCP-Server zugewiesen bekommen hat. Hier werden der Hostname von jedem Gerät und die zugehörige IP-Adresse angezeigt. Der Hostname des EUG 100 ist die Seriennummer des jeweiligen Gerätes.

Nachdem das Programm gestartet ist, wird der Benutzer aufgefordert, eine beliebige Taste zu drücken. Geschieht dies, wird die Funktion „WSAStartup“ aufgerufen, welche für die Verwendung von Sockets erforderlich ist. Ebenfalls wird ein Socket „s“ generiert:

```
s=socket(AF_INET,SOCK_STREAM,0);
```

Der erste Parameter beschreibt die Adressfamilie. Da TCP/IP benutzt wird, ist dies „AF_INET“. Da als Socket-Typ ein TCP-Socket erzeugt werden soll, wird „SOCK_STREAM“ als zweiter Parameter übergeben. An dritter Stelle steht das zu verwendende Protokoll: Steht an dieser Stelle eine Null, wird automatisch ein passendes Protokoll gewählt.

Ist das Socket erfolgreich erstellt, wird der Benutzer aufgefordert, die IP-Adresse des Servers (des Ethernet-UART-Gateways) einzugeben.

Jetzt werden der Struktur „SOCKADDR_IN“ über die Variable „addr“ folgende Werte übergeben:

```
addr.sin_family=AF_INET;
addr.sin_port=htons(1000);
addr.sin_addr.s_addr=inet_addr(IP);
```

Wie bereits bei der Erstellung des Sockets wird hier wieder die Adressfamilie übergeben. Der Port (das Ethernet-UART-Gateway benutzt Port 1000) und die vorher eingegebene IP-Adresse werden zugewiesen.

Da wir hier TCP/IP benutzen, wird die Struktur „SOCKADDR_IN“ verwendet. Diese ist jedoch mit der für den folgenden Befehl (connect) verwendeten Struktur „SOCKADDR“ kompatibel.

Soll nicht die IP-Adresse eingegeben werden, sondern der Hostname, gibt es die Funktion „gethostbyname“. Dies ist eine Hilfsfunktion, um bei bekannten Hostnamen die IP-Adresse zu ermitteln.

Jetzt kann sich das Client-Programm mit dem Server (dem Ethernet-UART-Gateway) über den erstellten Socket verbinden (connect):

```
check=connect(s,(SOCKADDR*)&addr,sizeof(SOCKADDR));
```

Hier werden zum einen das erstellte Socket „s“ und zum anderen ein Pointer auf die vorher mit Informationen gefüllte Struktur „SOCKADDR_IN“ angegeben. Als dritter Parameter wird die Größe dieser Struktur durch den „sizeof“-Operator übergeben.

Akzeptiert das Ethernet-UART-Gateway als Server daraufhin diese Verbindungsanfrage des Clients, kann mit der eigentlichen Datenübertragung begonnen werden. Sollte der Server nicht aktiv sein, wird „SOCKET_ERROR“ als Rückgabewert der Funktion „connect“ an die Variable „check“ ausgegeben. In diesem Fall erfolgt die Ausgabe einer Fehlermeldung und das Programm wird beendet. Im fehlerfreien Fall kann der Benutzer wählen:

Zum einen lassen sich Daten über das Netzwerk zum Ethernet-UART-Gateway versenden. Hierzu dient der folgende Quellcode:

```
printf(„\nZeichenfolge eingeben [max 1024]: “);
gets(buf);
send(s,buf,strlen(buf),0);
printf(„\nAnzahl gesendeter Bytes: %d\n“,strlen(buf));
```

Nach einer Textausgabe, die den Benutzer auffordert, eine Zeichenfolge einzugeben, werden alle Tastatureingaben in die Datei „buf“ gespeichert. Durch Abschluss der Eingabe durch die „Enter-Taste“ wird der Inhalt der char-Variablen „buf“ mit dem „send-Befehl“ zum Server gesendet. Hierbei wird das generierte Socket „s“ angegeben, „buf“ ist der bereits erwähnte Buffer und mit „strlen(buf)“ wird der Funktion die Länge der Zeichenkette übergeben. Das vierte Argument steht für optionale Spezifikationen für die Ausführung und wird als „flag“ bezeichnet (der Standardwert ist „0“). Dieser Quellcode erzeugt die in Abbildung 5 dargestellte Bildschirmausgabe.

Weiterhin kann der Benutzer des Client-Programms die möglicherweise bereits empfangenen Daten abrufen bzw. auf noch eingehende warten. Dieses geschieht mit der Funktion „recv“. Hierzu folgender Auszug aus dem Quelltext:

```
check=recv(s,buf,1024,0);
buf[check]='\0';
printf(„%s“,buf);
printf(„\n\nAnzahl empfangener Bytes: %d\n“,strlen(buf));
```

Die Parameter der Funktion „recv“ sind der der Funktion „send“ ähnlich. Das „s“ ist wieder das generierte Socket und die an vierter Stelle stehende „0“ das „flag“. Der zweite Übergabeparameter (buf) ist hierbei ein Pointer zu einem

```
Daten senden: 1      empfangene Daten abrufen: 2      (beenden: 0)
Ihre Wahl: 1

Zeichenfolge eingeben [max 1024]: 135987fgr
Anzahl gesendeter Bytes: 9
```

Bild 5: Daten zum Server senden (9 Byte (in diesem Fall: 135987fgr))

Buffer, in dem die empfangenen Daten gespeichert werden sollen. Die „1024“ kennzeichnet die Größe dieses Buffers. Die Abbildung 6 zeigt eine mögliche Bildschirmausgabe, hier sind 285 Bytes vom Ethernet-UART-Gateway empfangen worden.

```
Daten senden: 1      empfangene Daten abrufen: 2      (beenden: 0)
Ihre Wahl: 2

*aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
abbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbcccccc
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc#
Anzahl empfangener Bytes: 285
```

Bild 6: Daten vom Server empfangen (285 Byte)

Konfiguration

Das Ethernet-UART-Gateway ist universell zur Steuerung unterschiedlichster Geräte einsetzbar. Dazu ist es nötig, Einstellungen oder Konfigurationen bezüglich der UART-Schnittstelle vornehmen zu können, da die zu steuernden Geräte unterschiedliche UART-Einstellungen haben.

Folgende Einstellungen der UART-Schnittstelle können vorgenommen werden:

- Geschwindigkeit (Baudrate)
- Anzahl der Parity-Bits
- Anzahl der Stoppbits

Die Anzahl der Datenbits ist festgelegt mit 8 Bit.

Die Einstellung lässt sich auf zwei Arten über die Netzwerkschnittstelle vornehmen, einmal mit Hilfe eines Web-Servers, welcher später beschrieben ist, und andererseits durch ein Konfigurationsprotokoll, das in einem Datenpaket über Port 1001 versendet wird.

Dieses Datenpaket besteht lediglich aus drei Bytes (drei ASCII-Zeichen, siehe Abbildung 7) und enthält die Information über die Baudrate, die Parität und die Anzahl der Stoppbits.

Zuvor ist es natürlich wieder nötig, den bereits beschriebenen Verbindungsvorgang über das Netzwerk durchzuführen, jedoch dieses Mal nicht über Port 1000, sondern über den „Konfigurationsport“ 1001.

Welche Bedeutung die einzelnen Ziffern haben, ist in Tabelle 1 dargestellt. Aus dem Datenpaket mit dem Inhalt „425“ resultiert demnach eine Konfigurationseinstellung mit einer Baudrate von 9600 Bit/s (4), einer ungeraden Parität (2) und 1,5 Stoppbits (5).

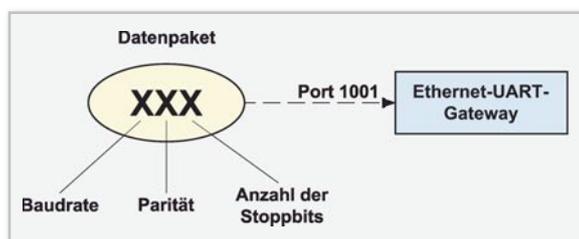


Bild 7: Aufbau des Protokolls zur Konfiguration über Port 1001

Baudrate in Bit/s	Wert im Protokoll	Parität	Wert im Protokoll	Anzahl Stoppbits	Wert im Protokoll
300	0	Gerade	1	1	1
1200	1	Ungerade	2	1,5	5
2400	2	Keine	0	2	2
4800	3	Markierung	3		
9600	4	Leerzeichen	4		
14400	5				
19200	6				
38400	7				
57600	8				
115200	9				

Die zweite Möglichkeit der Konfiguration bietet der integrierte, passwortgesicherte Web-Server.

Der Web-Server ist wie eine „gewöhnliche“ Internetseite mit einem normalen Web-Browser (z. B. Microsoft Internet Explorer oder Mozilla Firefox) aufrufbar.

Nach Eingabe des Hostnamens bzw. der Seriennummer (oder der IP-Adresse) in die Adresszeile des Web-Browsers wird der Benutzer durch das in Abbildung 8 dargestellte Fenster aufgefordert, den Benutzernamen (immer „admin“) und das Kennwort (Standardkennwort: „1111“) einzugeben.



Bild 8: Aufforderung zur Eingabe des Benutzernamens und des Kennworts

Bei korrekter Eingabe erscheint die in Abbildung 9 dargestellte Startseite des Ethernet-UART-Gateways.

Durch einen Mausklick auf die Schaltfläche „Konfiguration“ öffnet sich eine neue Seite (Abbildung 10). Hier ist jetzt die UART-Schnittstelle einzustellen. Durch sogenannte Auswahllisten ist es hier möglich, eine Auswahl bezüglich der Konfiguration zu treffen. Bei einem Klick auf die Schaltfläche „Einstellungen übertragen“ werden die Einstellungen dem Mikrocontroller übergeben.

Im Hauptfenster besteht weiterhin die Möglichkeit, das Kennwort, welches der Benutzer gleich nach Eingabe des Hostnamens/der IP-Adresse eingeben muss, zu ändern (Schaltfläche „Passwort ändern“). Der Screenshot dieser Seite ist in Abbildung 11 zu sehen. Hier kann der Benutzer jetzt – nachdem er als „admin“ eingeloggt ist – ein neues Passwort eingeben und es mit der Schaltfläche „Passwort ändern“ an den Mikrocontroller übertragen.

Schaltungsbeschreibung

Ethernetfähige Schaltungen bestehen in der Regel aus zwei Funktionseinheiten:

1. dem MAC (Media Access Controller) und
2. dem PHY (PHY steht für physikalische Schnittstelle).

In den meisten Fällen ist nur der MAC in der MCU zu finden (auch bei dem hier eingesetzten Mikrocontroller). Ein PHY muss in diesem Fall extern hinzugefügt werden.

Zu den Aufgaben eines Ethernet-MACs gehören in erster Linie die Paketbildung, der CSMA/CD-Buszugriff sowie grundlegende Protokollfunktionen. Der PHY bildet das physikalische Interface zum LAN-Kabel, also das Bindeglied zwischen Ethernet-MAC und dem LAN. Der PHY wird auch als Netzwerk-Transceiver bezeichnet. Abbildung 12 zeigt die Gesamtschaltung des EUG 100.

Mikrocontroller

Der für das Projekt eingesetzte Mikrocontroller ist der ATMEL AT91SAM7X256 (IC 1), eine auf ARM 7 basierende 32-Bit-MCU, der mit einem 18,432-MHz-Quarz (Q 1) getaktet und wie folgend beschrieben beschaltet wird. Die UART-Schnittstelle ist mit den beiden Datenleitungen (RxD und TxD) sowie den Steuerleitungen CTS und RTS auf eine Stiftleiste (ST 1) herausgeführt. Zwischen Stiftleiste und MCU sind jeweils 470- Ω -Widerstände bestückt. Sie dienen als Schutzbeschaltung für den Mikrocontroller.

Weiterhin ist die JTAG-Schnittstelle auf eine Stiftleiste (ST 2) herausgeführt. Über die JTAG-Schnittstelle ist der Mikrocontroller während der Entwicklungsphase programmiert worden, dementsprechend wird diese Stiftleiste in der ausgelieferten Baugruppe nicht mehr benötigt.

Netzwerk-Transceiver

IC 2 ist ein Netzwerk-Transceiver, der sowohl 10Base-T- sowie auch 100Base-T-Netzwerke (10 bzw. 100 MBit/s) unterstützt. Dieser Baustein sorgt für die Kommunikation zwischen dem Mikrocontroller und dem Netzwerk. Welche Netzwerkgeschwindigkeit vorliegt, wird über eine grüne SMD-LED auf der Platine angezeigt (D 2, 100 MBit/s: LED an). Weiterhin wird mit einer gelben SMD-LED (D 1) angezeigt, ob eine Full-Duplex-Verbindung (LED an) oder ob eine Half-Duplex-Verbindung besteht (LED aus). Die rote LED (D 3) zeigt die Netzwerkaktivität an (Link/Act). Das Netzwerk-Transceiver-IC wird via Q 2, einen 25-MHz-Quarz, mit einem Taktsignal versorgt. Die Betriebsspannung wird dem IC über drei Pins zugeführt (Pin 23, 30 und 41), die für die Netzwerkbuchse benötigte Ausgangsspannung wird ebenfalls über drei Pins „nach außen“ geführt (Pin 1, 2 und 9). Diese Spannungs-Ein- bzw. -Ausgänge sind mit je zwei Koppelkondensatoren (C 25 bis C 34, 10 nF und 100 nF) beschaltet, die jeweils so nahe wie möglich an den zugehörigen Pins platziert sind.

Spannungsversorgung

Der Mikrocontroller sowie der Netzwerk-Transceiver benötigen eine Betriebsspannung von 3,3 V. Die Spannung wird über einen „Step-down-Regler“ (IC 3) bereitgestellt. Die Eingangsspannung, welche über eine Stiftleiste (siehe Abbildung 2) angeschlossen wird, muss im Bereich von 5 V bis 40 V liegen.



Bild 9: Startseite des Web-Servers (Internet Explorer)



Bild 10: Konfigurationsseite des Web-Servers (Internet Explorer)

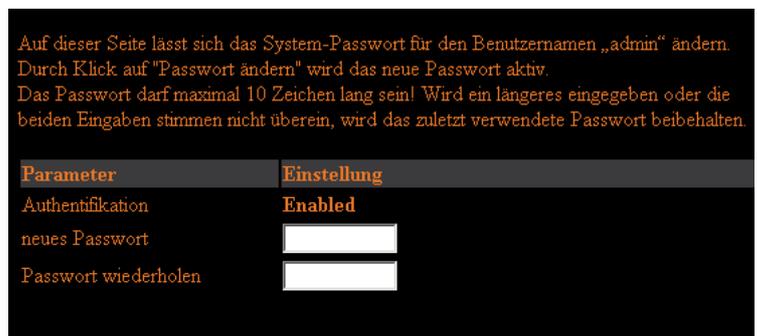


Bild 11: Passwortseite des Web-Servers (Internet Explorer)

Folgender Hinweis ist hierbei zu beachten:

Zur Gewährleistung der elektrischen Sicherheit muss es sich bei der speisenden Quelle um eine Sicherheits-Schutzkleinspannung handeln. Außerdem muss es sich um eine Quelle begrenzter Leistung handeln, die nicht mehr als 15 W Leistung liefern kann. Diese Forderung ist jedoch meist leicht zu erfüllen, z. B. durch Einsatz eines 12-V-Steckernetzteils mit max. 500 mA Strombelastbarkeit (6 W).

Die Spule L 1, Diode D 5, Kondensator C 9 sowie die Widerstände R 23 und R 26 sind Applikations-Vorgaben aus dem Datenblatt des Reglers. Die Ausgangsspannung wird durch die Widerstände R 24, R 27 und R 31 bestimmt. Die Kondensatoren C 7, C 8, C 10, C 11 und C 12 dienen als Abblock- bzw. als Stützkondensatoren.

Nachbau

Der Nachbau der Schaltung ist unkompliziert, da alle SMD-Bauteile bereits vorbestückt sind. Diese Bestückung ist lediglich noch auf Fehler zu kontrollieren.

Der weitere Aufbau erfolgt unter Zuhilfenahme der Stückliste, des Bestückungsplans und der Platinenfotos.

Bei den Elektrolytkondensatoren (C 7, C 10, C 51) sowie bei

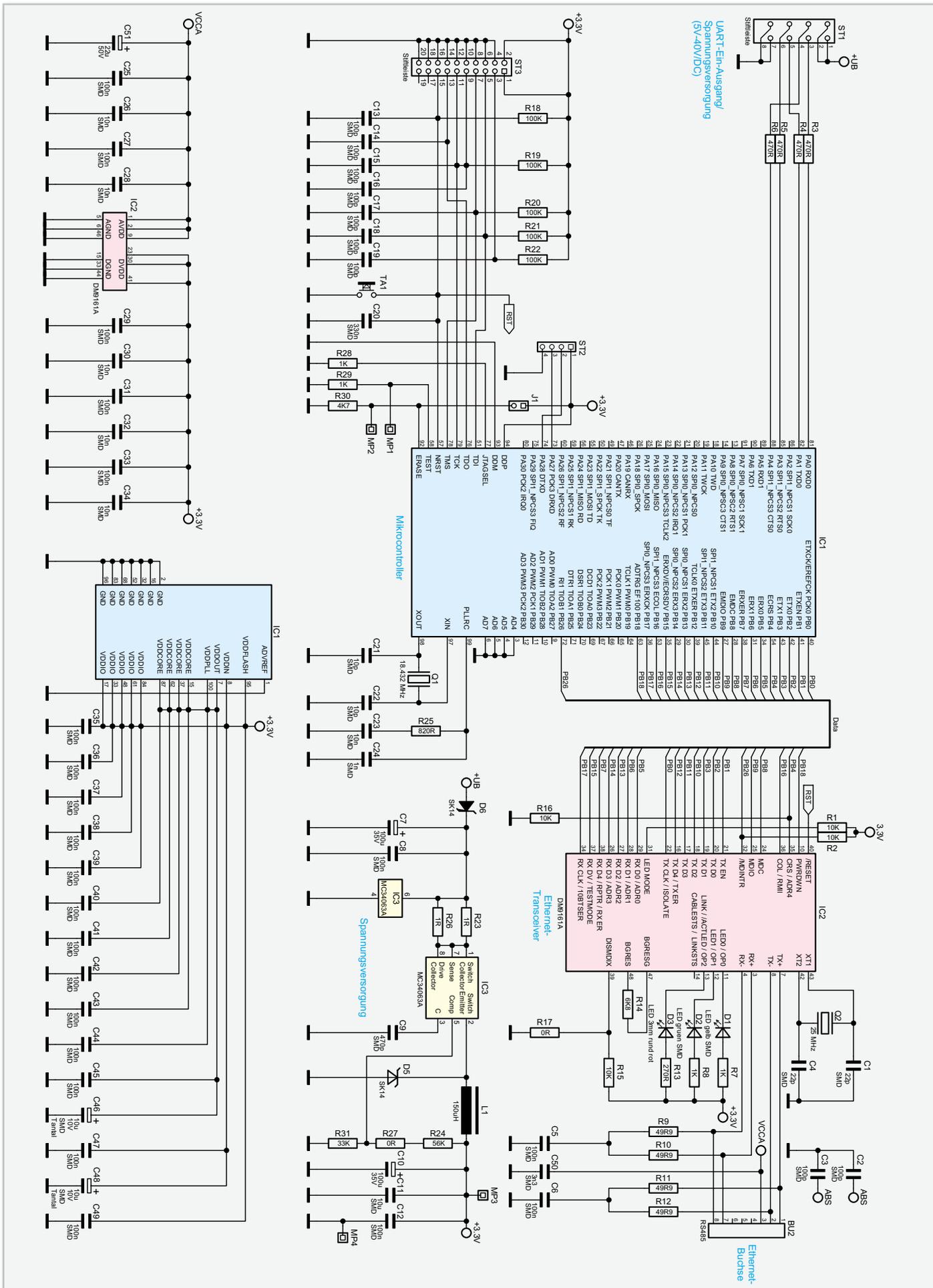
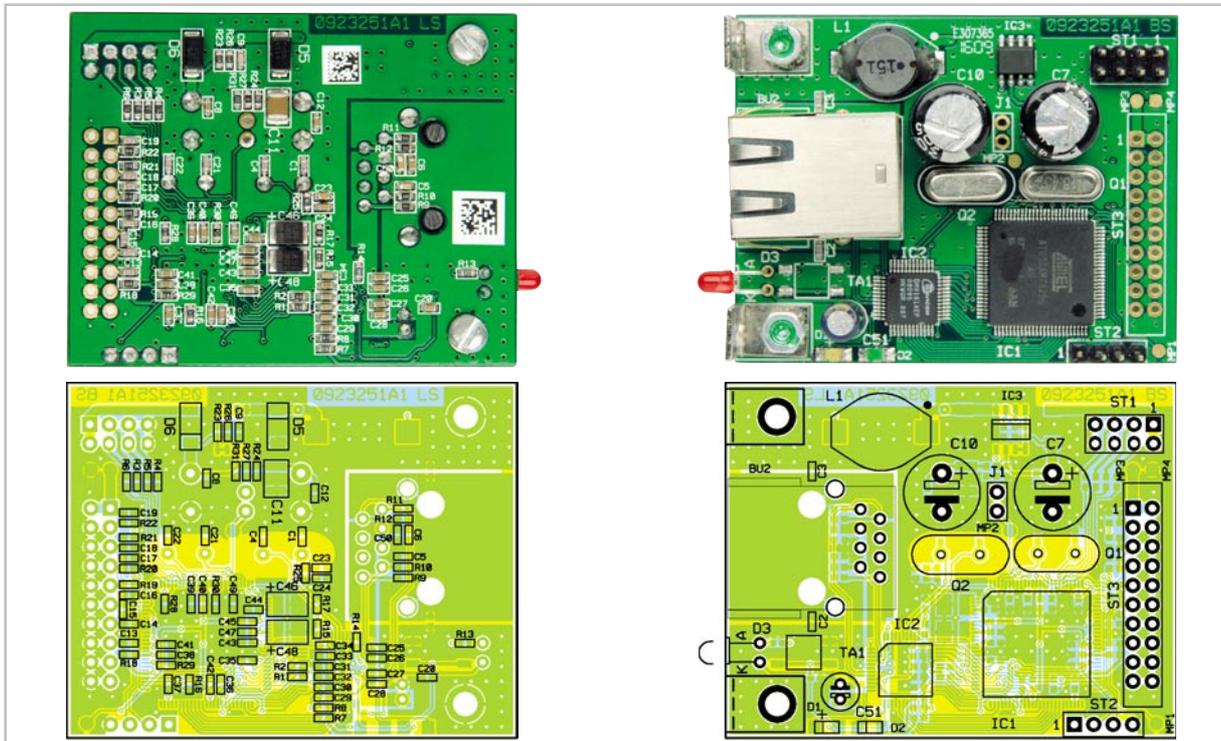


Bild 12: Das Schaltbild des Ethernet-UART-Gateway

der LED (D 3) ist auf die richtige Polung beim Einbau zu achten. Die LED-Anschlüsse sind vor dem Einbau abzuwinkeln, so dass die LED in eine evtl. vorhandene Abdeckplatte passt. Ansonsten sind die Stiftleiste ST 1, die beiden Quarze Q 1 bzw. Q 2 und die Netzwerkbuchse BU 2 zu bestücken. Abschließend besteht die Möglichkeit, zwei kleine Metallwinkel für eine Montage an eine Abdeckplatte zu befestigen.

Für die platinenseitige Befestigung sind zwei M3-Schrauben sowie passende Muttern und Fächerscheiben vorgesehen. Vor deren Festziehen ist darauf zu achten, dass die Winkel bündig mit dem Platinenrand abschließen. **ELV**

Link: [1] www.elvjournal.de



Ansicht der fertig bestückten Platine des EUG 100, rechts die Oberseite, mit jeweils zugehörigem Bestückungsplan und links die Unterseite (SMD).

Stückliste: EUG 100

Widerstände:

0 Ω /SMD/0603	R17, R27
1 Ω /SMD/0603	R23, R26
49,9 Ω /SMD/0603	R9–R12
270 Ω /SMD/0603	R13
470 Ω /SMD/0603	R3–R6
820 Ω /SMD/0603	R25
1 k Ω /SMD/0603	R7, R8, R28, R29
4,7 k Ω /SMD/0603	R30
6,8 k Ω /SMD/0603	R14
10 k Ω /SMD/0603	R1, R2, R15, R16
33 k Ω /SMD/0603	R31
56 k Ω /SMD/0603	R24
100 k Ω /SMD/0603	R18–R22

Kondensatoren:

10 pF/SMD/0603	C21, C22
22 pF/SMD/0603	C1, C4
100 pF/SMD/0603	C2, C3, C13–C19
470 pF/SMD/0603	C9
1 nF/SMD/0603	C24
3,3 nF/SMD/0603	C50
10 nF/SMD/0603	C23, C26, C28, C30, C32, C34
100 nF/SMD/0603	C5, C6, C8, C12, C25, C27, C29, C31, C33, C35–C45, C47, C49

330 nF/SMD/0603	C20
10 μ F/SMD/1210	C11
10 μ F/10 V/SMD/tantal	C46, C48
22 μ F/50 V/105 $^{\circ}$ C	C51
100 μ F/35 V	C7, C10

Halbleiter:

ELV08862/SMD	IC1
DM9161AEP/SMD	IC2
MC34063AD/SMD	IC3
SK14/SMD	D5, D6
LED, SMD, Gelb, low current	D1
LED, SMD, Grün, low current	D2
LED, 3 mm, Rot	D3

Sonstiges:

Quarz, 18,432 MHz, HC49U4	Q1
Quarz, 25 MHz, HC49U	Q2
SMD-Induktivität, 150 μ H/1 A	L1
Modulare Einbaubuchse RJ45-248TC1, 8-polig, print, abgeschirmt	BU2
Stiftleiste, 2x 4-polig, gerade, print	ST1
Stiftleiste, 1x 4-polig, gerade, print	ST2
2 Zylinderkopfschrauben, M3 x 6 mm	
2 Muttern, M3	
2 Fächerscheiben, M3	
2 Befestigungswinkel, vernickelt	