

## Teil 2

# USB-Stick-Interface STI 100

So günstig und mit reichlich Speicherkapazität sie erhältlich sind, im Gegensatz zu anderen Speichermedien ist die Integration von USB-Sticks in eigene Schaltungsapplikationen ein kaum zu überwindendes Hindernis. Das Interface STI 100 bietet die Möglichkeit, einen USB-Stick einfach per UART- oder SPI-Schnittstelle anzusprechen. Der zugehörige Befehlssatz unterstützt vielfältige Dateioperationen, so dass die Verarbeitung von gesammelten Messwerten genauso möglich ist wie die Nutzung von auf einem PC erzeugten Daten (z. B. Konfigurationen) in der eigenen Schaltung. Im zweiten Teil besprechen wir die Schaltungstechnik, die Inbetriebnahme sowie die Anwendung des Befehlssatzes.

## Schaltung

Abbildung 3 zeigt die Schaltung des Interface-Bausteins. Der Anschluss des USB-Stick-Interfaces erfolgt über die Stiftleiste ST 1. Die 5-V-Versorgungsspannung ist dabei an Pin 1 anzuschließen und Pin 2 ist mit Masse zu verbinden. Die gesamte Pin-Belegung der Stiftleiste kann man Tabelle 5 entnehmen, wobei die Funktion der Pins 3 bis 6 von der gewählten Schnittstelle abhängig ist.

Das Interface kann sowohl über die UART- als auch die SPI-Schnittstelle angesprochen werden. Auf die parallele Schnittstelle haben wir aus Platzgründen verzichtet. Die Auswahl der Schnittstelle erfolgt entsprechend Tabelle 6 über den Jumper J 1. Für den regulären Betrieb muss der Jumper J 3 auf Position „Normal“ gesteckt werden. Befindet sich J 3 in der Stellung „Prog“ und wird nun ein Reset durchgeführt,

gelangt IC 1 in den Programmier-Modus. Nun ist über die UART-Schnittstelle ein Update der Firmware durchführbar. Dazu ist allerdings ein spezieller Programmieradapter not-

**Tabelle 5: Pin-Belegung der Stiftleiste ST 1**

| Pin ST 1 | Beschreibung   |                     |              |           |
|----------|----------------|---------------------|--------------|-----------|
| 1        | +5 V           |                     |              |           |
| 2        | GND            |                     |              |           |
|          | UART-Interface | SPI-Slave-Interface | Pin Nr. IC 1 | Name IC 1 |
| 3        | /CTS           | CS                  | 34           | ADBUS3    |
| 4        | /RTS           | SDO                 | 33           | ADBUS2    |
| 5        | RXD            | SDI                 | 32           | ADBUS1    |
| 6        | TXD            | SCLK                | 31           | ADBUS0    |

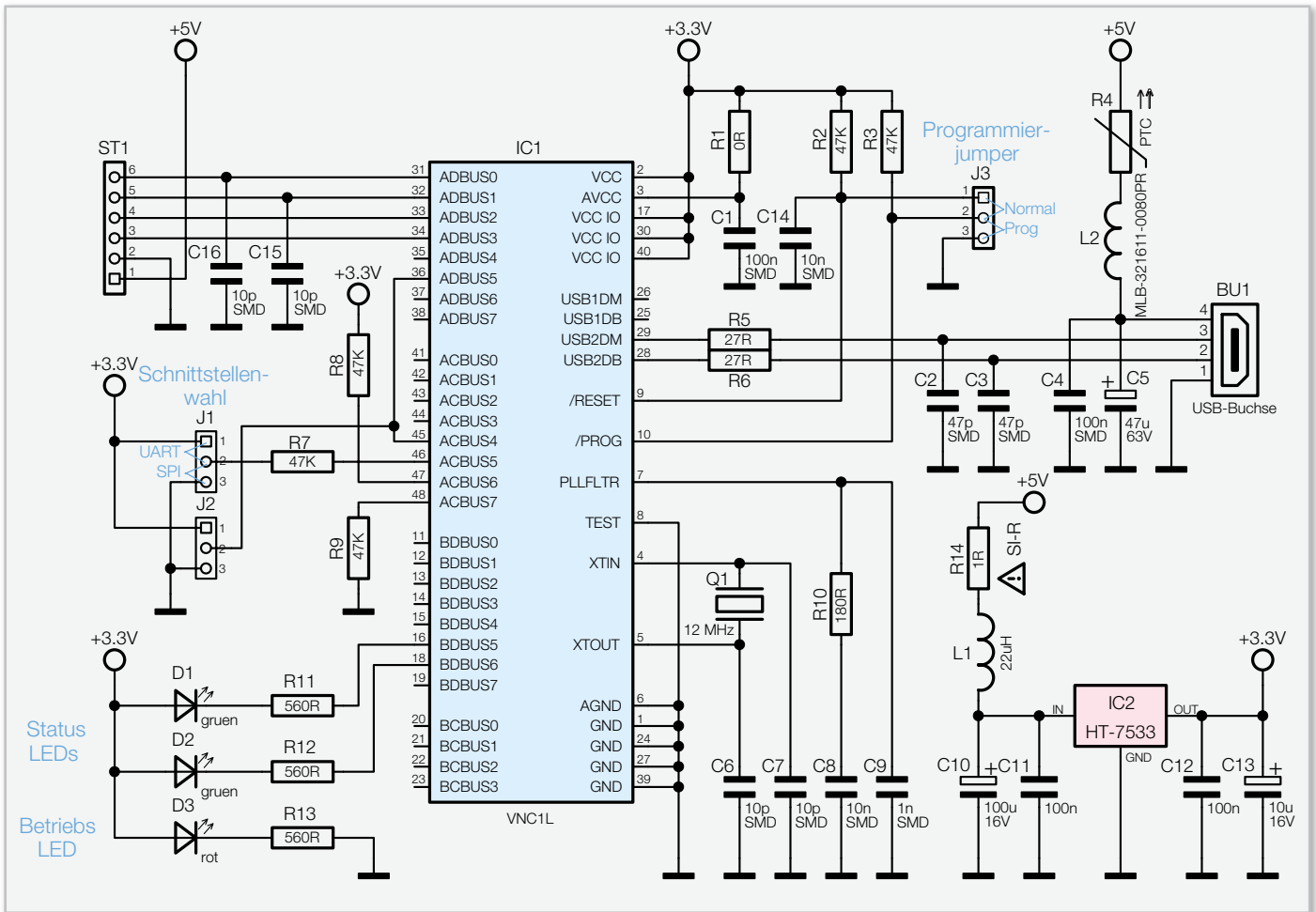


Bild 3: Schaltbild des ST1 100

Tabelle 6: Die Auswahl der Schnittstelle

| ACBUS5 (Pin 46) | Schnittstelle |
|-----------------|---------------|
|                 | SPI           |
|                 | UART          |

wendig. Nähere Informationen hierzu findet man unter der Internet-Adresse [1].

Das Herzstück der Schaltung ist der USB-Host-Controller „Vinculum VNC1L“. Wie bereits im ersten Teil beschrieben, verwaltet IC 1 das Dateisystem des USB-Sticks und stellt eine einfache Softwareschnittstelle zur Verfügung.

Zur Takterzeugung muss der interne Oszillator mit einem 12-MHz-Quarz (Q 1) beschaltet werden. Ein ebenfalls integrierter PLL-Block erzeugt daraus 24- und 48-MHz-Taktsignale für die internen Funktionsblöcke.

Die notwendige Versorgungsspannung (3,3 V) wird von IC 2 zur Verfügung gestellt. Dementsprechend haben die Ausgangsspannungen des VNC1L einen Pegel von 3,3 V. Sollte man das Interface an eine Schaltung mit abweichender Versorgungsspannung anschließen wollen, sind die Pegel entsprechend anzupassen. Die Eingangspins akzeptieren Spannungspegel bis 5 V.

LED D 3 dient als Betriebsanzeige-LED, sie leuchtet, sobald die Versorgungsspannung angelegt wird. Mit den beiden

LEDs D 1 und D 2 signalisiert der VNC1L den USB-Zugriff. Die beiden SMD-Spulen L 1 und L 2 dienen der Störsignalunterdrückung und sollen verhindern, dass Störsignale in die Ziel-Hardware gelangen.

R 4 ist ein PTC-Baustein und realisiert eine reversible Kurzschlussicherung, die die Schaltung bei einem Defekt des USB-Sticks schützen soll.

## UART-Schnittstelle

Standardmäßig sind folgende Parameter für die UART-Schnittstelle eingestellt:

9600 Baud, 8 Datenbits, 1 Startbit, 1 Stoppbit, keine Parität

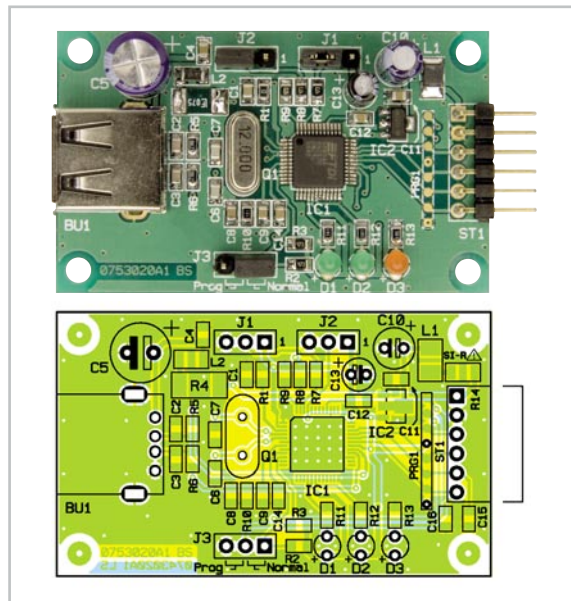
Die Baudrate kann jederzeit per Befehl geändert werden, wobei der entsprechende Baudratendivisor der Tabelle 3 („ELV-Journal“ 5/2007) zu entnehmen ist. Der VNC1L erwartet ein Hardware-Handshake über die Signalleitungen /RTS und /CTS (ST 1, Pin 3 und Pin 4).

Möchte man ohne Handshake auskommen oder ist ein Handshake nicht möglich, kann der /CTS-Pin mit Masse verbunden werden.

Es ist darauf zu achten, dass die Anschlussleitungen zwischen dem ST1 100 und der angeschlossenen Schaltung nicht länger als max. 3 m sind.

## Nachbau

Da bereits alle SMD-Bauteile vorbestückt sind, beschränkt sich der Nachbau auf das Bestücken der bedrahteten Bauteile. Wie gewohnt, erfolgt die Bestückung anhand des Bestückungsplans, der Stückliste und unter Zuhilfenahme der Platinenfotos. Zuerst werden die Stiftleisten in die vorgesehenen Bohrungen gesteckt und auf der Lötseite verlötet. Dann folgt die USB-Buchse, wobei darauf zu achten ist, dass deren Gehäuse direkt auf der Leiterplatte aufliegt, so dass die mechanische Beanspruchung der Lötstellen so gering wie möglich ist. Als Letztes werden die Elektrolyt-Kondensatoren und die Leuchtdioden bestückt. Hier ist auf die richtige Polung zu achten. Elkos sind dabei üblicherweise am Minuspol durch eine Gehäusemarkierung gekennzeichnet. Die Katode einer LED ist durch den jeweils kürzeren Anschluss zu erkennen. Zum Schluss sind nur noch die Jumper aufzustecken.



Ansicht der fertig bestückten Platine des STI 100

## Inbetriebnahme

Die Inbetriebnahme erfolgt beispielhaft für die UART-Schnittstelle. Sinngemäß kann man die Vorgehensweise für die SPI-Schnittstelle übernehmen, wobei jedoch die typischen Eigenschaften der Schnittstellen zu beachten sind. Zu Beginn sollte die Schaltung nochmals auf Bestückungsfehler oder Lötzinnbrücken überprüft werden. Über den Jumper J 1 wird jetzt die gewünschte Schnittstelle (Tabelle 6) ausgewählt und JP 3 auf „Normal“ gesteckt. J 2 ist grundsätzlich so zu stecken, dass Pin 36 und Pin 45 des VNC1L auf Versorgungsspannungspotential liegen. Die zweite Jumperstellung von J 2 bleibt in dieser Applikation ungenutzt. Danach sollte die Schaltung an ein Labornetzteil angeschlossen werden, um die Stromaufnahme zu prüfen. Sie sollte die in den technischen Daten genannte Angabe nicht überschreiten. Nun kann das Interface mit der Ziel-Hardware verbunden werden, die Belegung der Stiftleiste ist Tabelle 5 zu entnehmen. Nach dem Anlegen der Versorgungsspannung gibt der VNC1L folgende Meldung aus:

**Ver xx.xxVcccf On-Line**

Dabei stellt xx.xx die Firmwareversion und Vcccf den Firmwaretyp dar.

Sobald ein USB-Stick angeschlossen wird, meldet IC 1:

**Device Detected P2**

Nun erfolgt die Initialisierung des USB-Sticks, dabei wird das Dateisystem gelesen und die freien Sektoren werden gezählt. Je nach Speicherkapazität und Dateisystem (FAT, FAT32) kann dies einige Sekunden dauern. Der angeschlossene USB-Stick zeigt dies durch Blinken der Aktivitäts-LED an. Zum Schluss wird im Hauptverzeichnis nach einer Firmware-Update-Datei gesucht. Ist ein Update vorhanden, wird die Firmware automatisch aktualisiert.

**Achtung!** Hierfür ist eine spezielle Firmware notwendig, die auf der Webseite des Herstellers [2] unter dem Link „Reflash (FTD)“ erhältlich ist.

## Stückliste: USB-Stick-Interface STI 100

### Widerstände:

|   |               |
|---|---------------|
| 0 $\Omega$ /SMD/0805                      | R1            |
| Sicherungswiderstand 1 $\Omega$ /SMD/1206 | R14           |
| 27 $\Omega$ /SMD/0805                     | R5, R6        |
| 180 $\Omega$ /SMD/0805                    | R10           |
| 560 $\Omega$ /SMD/0805                    | R11–R13       |
| 47 k $\Omega$ /SMD/0805                   | R2, R3, R7–R9 |
| Polyswitch, 0,75 A, SMD, 1812             | R4            |

### Kondensatoren:

|                  |                  |
|------------------|------------------|
| 10 pF/SMD/0805   | C6, C7, C15, C16 |
| 47 pF/SMD/0805   | C2, C3           |
| 1 nF/SMD/0805    | C9               |
| 10 nF/SMD/0805   | C8, C14          |
| 100 nF/SMD/0805  | C1, C4, C11, C12 |
| 10 $\mu$ F/16 V  | C13              |
| 47 $\mu$ F/63 V  | C5               |
| 100 $\mu$ F/16 V | C10              |

### Halbleiter:

|                      |        |
|----------------------|--------|
| ELV07692/SMD (VNC1L) | IC1    |
| HT7533/SMD           | IC2    |
| LED, 3 mm, Grün      | D1, D2 |
| LED, 3 mm, Rot       | D3     |

### Sonstiges:

|  |       |
|--|-------|
| Quarz, 12 MHz, HC49U4                      | Q1    |
| SMD-Induktivität, 22 $\mu$ H, 250 mA       | L1    |
| Chip-Ferrit, 1206, 80 $\Omega$ bei 100 MHz | L2    |
| USB-A-Buchse, winkelprint, liegend         | BU1   |
| Stiftleiste, 1 x 6-polig, winkelprint      | ST1   |
| Stiftleiste, 1 x 3-polig, gerade, print    | J1–J3 |
| Jumper, geschlossene Ausführung            | J1–J3 |

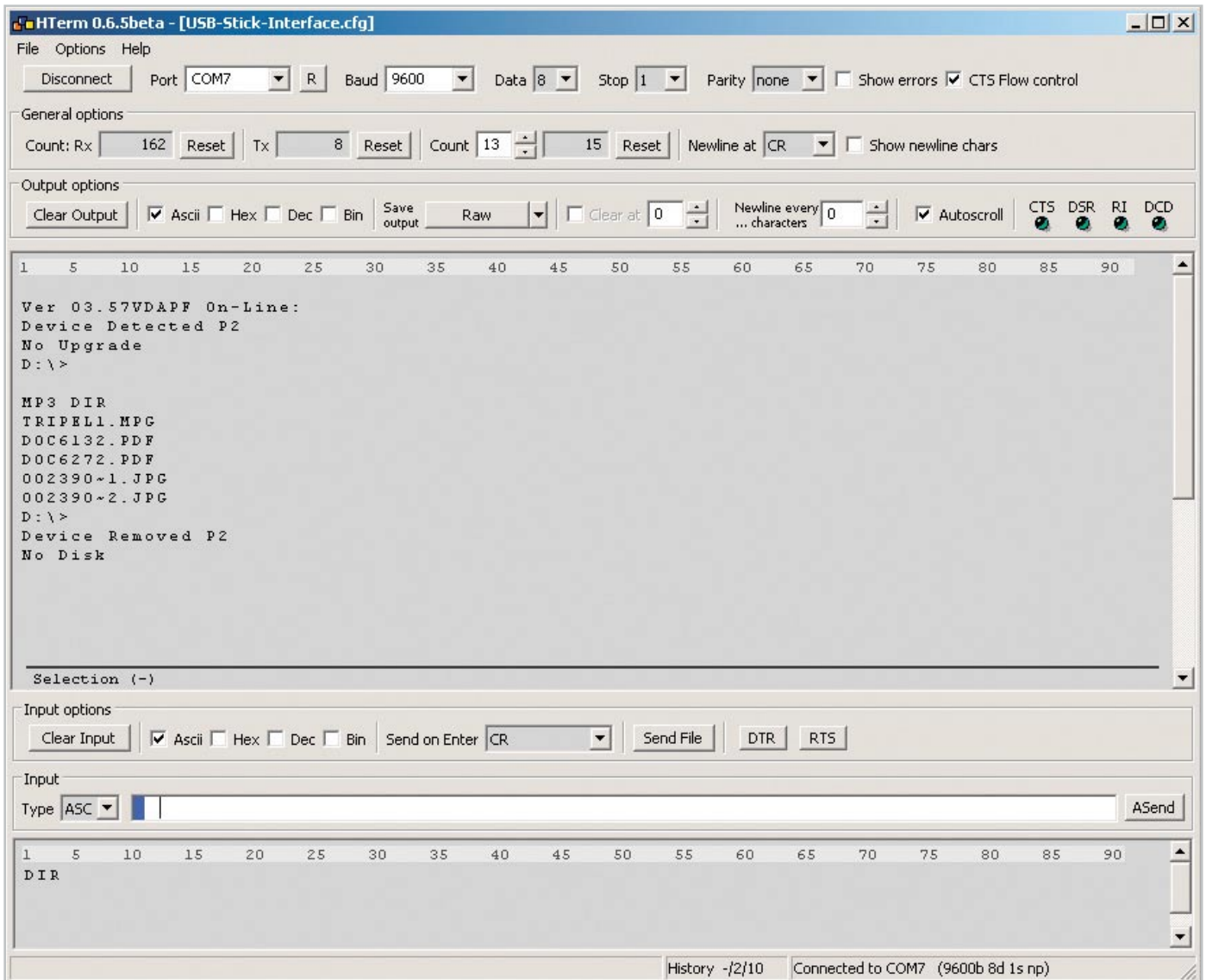


Bild 4: Kommunikation mit dem STI 100 über UART mit HTerm

Wird keine Update-Datei gefunden, meldet der VNC1L:

### No Upgrade

D:\>

Jetzt kann der VNC1L mit den vorhandenen Befehlen konfiguriert oder Daten können vom USB-Stick gelesen oder geschrieben werden.

Für die ersten Experimentierschritte kann man das Interface auch über ein Terminalprogramm (z. B. HTerm [3]) auf dem PC bedienen. Dazu ist das STI 100 an die serielle Schnittstelle des PCs anzuschließen. Für die notwendige Anpassung der Spannungspegel kann ein USB/UART- oder RS232/UART-Umsetzer zwischen PC und Interface geschaltet werden. Das Terminalprogramm wird entsprechend den Angaben im Abschnitt UART-Schnittstelle konfiguriert und die Schnittstelle geöffnet. Man kann nun alle Befehle ausprobieren und deren Anwendung erproben.

Abbildung 4 zeigt die Meldungen nach dem Anschließen der Versorgungsspannung und die Antwort auf den Befehl „DIR“. Die Firmware trägt die Versionsnummer 3.57 und es wurde keine Update-Datei auf dem angeschlossenen USB-Stick entdeckt. Auf dem USB-Stick befinden sich 5 Dateien sowie der Ordner MP3. Die beiden letzten Zeilen zeigen an, dass der USB-Stick entfernt wurde.

Die Verwendung des STI 100 im Zusammenspiel mit einem Mikrocontroller ist prinzipiell identisch. Als Erstes muss die UART-Schnittstelle des Controllers konfiguriert werden, danach kann der Controller auf den VNC1L bzw. auf den USB-Stick zugreifen.

In Abbildung 5 ist anhand eines kurzen Beispielquellcodes die Handhabung des STI 100 auf Mikrocontroller-Ebene dargestellt. Der Auszug zeigt die Initialisierung des VNC1L, dabei wird auf den Mikrocontroller-Befehlssatz umgeschaltet und die Baudrate auf 1.000.000 eingestellt. Der vollständige Quellcode kann auf der ELV-Produktseite heruntergeladen werden. Das Projekt bietet einfache Funktionen für das Lesen, Schreiben, Erstellen und Löschen einer Datei. **ELV**

### Internet:

- [1] <http://www.vinculum.com>
- [2] <http://www.vinculum.com/downloads.html#vfirmware>
- [3] <http://www.der-hammer.info/terminal>

```

void main()
{
    .
    .
    .
    uart_init(); // Initialisiert die UART-Schnittstelle des Controllers
    if( vnc1l_init() < 3 ) // STI 100 initialisieren und Rückgabewert auswerten
    {
    }
    else
    {
    }

    .
    .
    .
}

void switch_to_short_instruction_set(void)
{
    ucRXI = WAIT_STATE; // Antwortvariable initialisieren
    ucSioPointer = 0; // Indexzeiger des Empfangspuffers löschen
    uart_putchar('S');
    uart_putchar('C');
    uart_putchar('S');
    uart_putchar(0x0d); // ASCII-Steuerzeichen CR
}

void set_vnc1l_baudrate(void) // Ändert die Baudrate des VNC1L auf 1.000.000
{
    ucRXI = WAIT_STATE; // Antwortvariable initialisieren
    ucSioPointer = 0; // Indexzeiger des Empfangspuffers löschen
    uart_putchar(0x14); // Befehl „SBD“
    uart_putchar(0x20); // ASCII-Zeichen Space
    uart_putchar(0x03); // 0x030000 entspricht einer Baudrate von 1.000.000
    uart_putchar(0x00);
    uart_putchar(0x00);
    uart_putchar(0x0d); // ASCII-Steuerzeichen CR
}

unsigned char vnc1l_init(void)
{
    unsigned char error = 0;

    do
    {
        switch_to_short_instruction_set(); // Auf Mikrocontroller-Befehlssatz umschalten
        wait_for_vnc1l(); // Antwort des VNC1L abwarten
        set_vnc1l_baudrate(); // Baudrate ändern
        wait_for_vnc1l(); // Antwort des VNC1L abwarten
        set_baudrate(); // Baudrate der UART-Schnittstelle des
        // Controllers ändern
        if(ucErrorCnt == 0) // Ist ein Fehler aufgetreten?
            error = 0;
        else
            error++;
    }while( (error != 0) && (error < 3) ); // Abbruch nach drei Versuchen

    return error;

    // Funktion wartet 500 ms auf die Antwort des VNC1L. Ist eine Antwort
    // empfangen worden, enthält ucRXA den Typ der Antwort
}

unsigned char wait_for_vnc1l(void)
{
    uiTimeOut = 50; // 500 ms Timeout
    while( (ucRXA == WAIT_STATE) && (uiTimeOut) );
    if(uiTimeOut == 0)
    {
        ucErrorCnt++; // Fehlervariable inkrementieren
        return 1;
    }
}

```

Bild 5: Quellcode zur Initialisierung des STI 100