



Dokument NR.: AB-001-U-01

ASCII ZEICHEN AUF DEM HYPERTERMINAL



Übungsaufgabe RS232 Schnittstelle



Bitte denken Sie an die Umwelt,
bevor Sie diese Datei ausdrucken

INHALTSVERZEICHNIS

1. Aufgabe 1	3
2. Aufgabe 2	4
3. Lösung Aufgabe 1.....	5
4. Lösung Aufgabe 2.....	7

BILDVERZEICHNIS

Bild 1. Aufgabe 1: Hyperterminal unter Windows XP.....	3
Bild 2. verwendete Boards SB-001 (weiß), AB-002 (schwarz).....	3
Bild 3. Aufgabe 2: Hyperterminal unter Windows XP.....	4

MODIFICATION HISTORY:

Version	Date	Comments
1.0	09.2012	first release

1. AUFGABE 1

Schreiben Sie ein Programm, das alle sichtbaren ASCII Zeichen auf den Hyperterminal ausgibt.
Folgende Punkte müssen beachtet werden:

1. Schnittstellenparameter: 9600Baud, 8 Datenbit, 1 Stopp-Bit, asynchroner Betrieb ohne Empfang
2. Timer 2
3. Benutzen Sie SBUF Register, keine printf() Funktion
4. Nur druckbare (sichtbare) Zeichen ausgeben

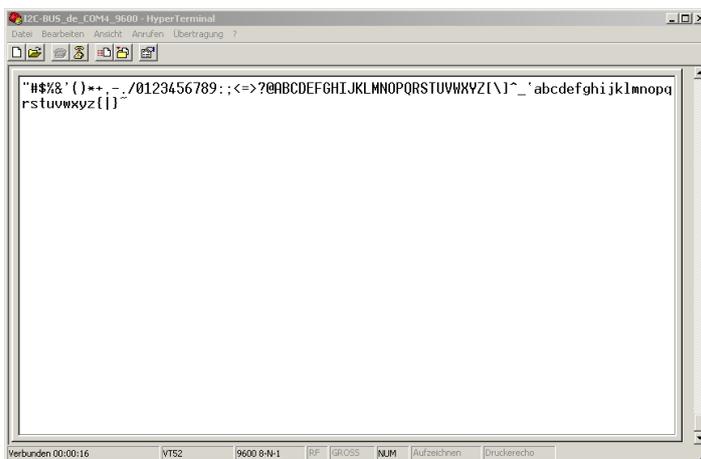


Bild 1. Aufgabe 1: Hyperterminal unter Windows XP

Für diese Aufgabe werden zwei Boards benötigt.

- Mikrocontrollerboard z.B. SB-001/SB-002
- Serielle Schnittstelle AB-001/AB-002

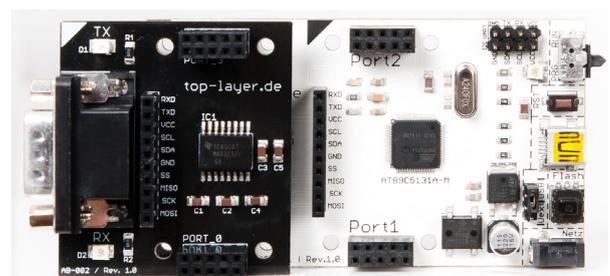
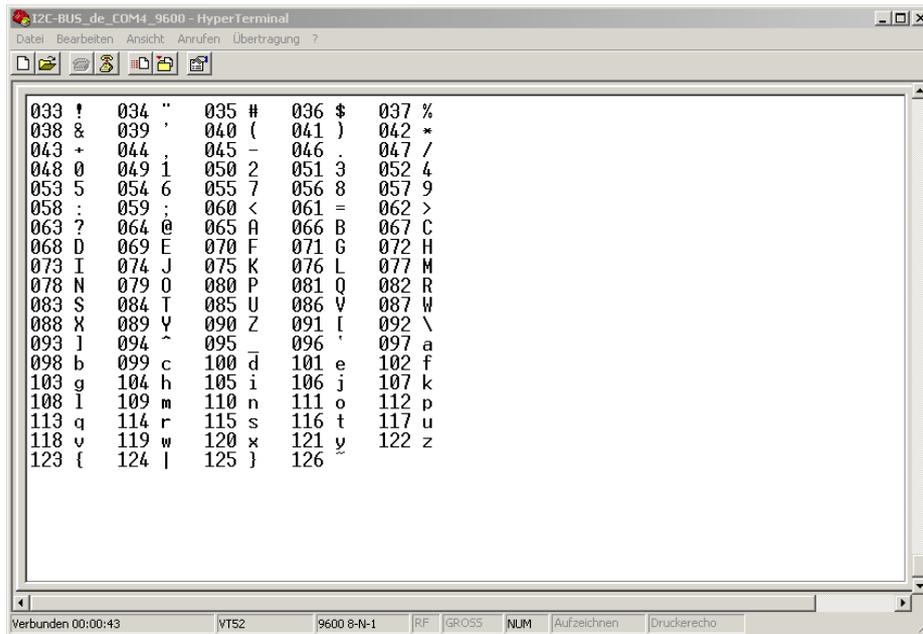


Bild 2. Verwendete Boards SB-001 (weiß), AB-002 (schwarz)

2. AUFGABE 2

Erweitern Sie die Aufgabe 1 um folgende Punkte:

1. Zu den ASCII Zeichen soll Dezimal-Wert ausgegeben werden.
2. Die Hyperterminal -Ausgabe soll wie auf dem Bild angepasst werden



```
i2c-BUS_de_COM4_9600 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
033 ! 034 " 035 # 036 $ 037 %
038 & 039 ' 040 ( 041 ) 042 *
043 + 044 , 045 - 046 . 047 /
048 0 049 1 050 2 051 3 052 4
053 5 054 6 055 7 056 8 057 9
058 : 059 ; 060 < 061 = 062 >
063 ? 064 @ 065 A 066 B 067 C
068 D 069 E 070 F 071 G 072 H
073 I 074 J 075 K 076 L 077 M
078 N 079 O 080 P 081 Q 082 R
083 S 084 T 085 U 086 V 087 W
088 X 089 Y 090 Z 091 [ 092 \
093 ] 094 ^ 095 _ 096 ` 097 a
098 b 099 c 100 d 101 e 102 f
103 g 104 h 105 i 106 j 107 k
108 l 109 m 110 n 111 o 112 p
113 q 114 r 115 s 116 t 117 u
118 v 119 w 120 x 121 y 122 z
123 { 124 | 125 } 126 ~
```

Verbunden 00:00:43 VT52 9600 8-N-1 RF GROSS NUM Aufzeichnen Druckerecho

Bild 3. Aufgabe 2: Hyperterminal unter Windows XP

3. LÖSUNG AUFGABE 1

```

/*****
Datei:      Aufgab1_AB-001-U-01_ACII_ZeichenHyperterminal.c

Datum:      Sep.2012 - first release
Version    1.0

Autor:      Viktor Schabelski info@i2c-bus.de
Lizenz:     Creative Commons Lizenz
            Namensnennung - Keine kommerzielle Nutzung - Keine Bearbeitungen
            www.i2c-bus.de/cc

Uebungsaufgabe:
Aufgabe 1
    Schreiben Sie ein Programm, das alle sichtbaren ACII Zeichen auf den Hyperterminal
    ausgibt.

    Folgende Punkte müssen beachtet werden:
        1. Schnittstellenparameter: 9600Baud, 8 Datenbit, 1 Stopp-Bit,
           asynchroner Betrieb ohne Empfang
        2. Timmer 2
        3. Benutzen Sie SBUF Register
        4. printf() Funktion darf nicht benutzt werden
        5. Nur druckbare (sichtbare) Zeichen ausgeben

    Fuer diese Aufgabe werden zwei Boards benoetigt.
        1. Mikrocontrollerboard z.B. SB-001/SB-002
        2. AB-001/AB-002 Serielle Schnittstelle

*****/

/*****
Includes
*****/
#include <at89c5131.h>
#include <stdio.h>

/*****
Globale Variablen
*****/
//Quarz
#define _24MHZ      24
#define _12MHZ     12

#define ON 1

/*****
Funktionsprototypen
*****/
void init_UART(char quarz);

```

```

/*****
Funktionen
*****/
void main (void){
    char myZeichen = 0x21;    // ab Zeichen"!
    init_UART(_24MHZ);

    do{
        SBUF = myZeichen;    // Zeichen in SBUF Register laden
        myZeichen++;        // Zeichen um 1 erhoen
        while(TI == 0);     // Warten bis Zeichen uebertragen ist
        TI = 0;             // TI (Transmit Interrupt) Flag im Register SCON loeschen
    }while(myZeichen <= 126); // bis Zeichen "~"

    while(1);
}

/*****
Initialisierung RS232/V24 Schnittstelle.

Initialisierung der seriellen Schnittstelle in Mode 1
Schnittstellenparameter: 9600Baud, 8 Datenbit, 1 Stopp-Bit,
asynchroner Betrieb ohne Empfang,
Quarz Frequenz 12/24 MHz
Timmer2
SCON = 0x40h, Serial Control Register (0x98h)
FE/SM0 - 0 |
SM1 - 1 |
SM2 - 0 |
REN - 0 |
TB8 - 0 |
RB8 - 0 |
TI - 0 |
RI - 0 |

T2CON = 0x30h, Timer 2 Control Register (0xC8h)
TF2 - 0 |
EXF2 - 0 |
RCKL - 1 |
TCKL - 1 |
EXEN2 - 0 |
TR2 - 0 |
C/T2# - 0 |
CP/RL2# - 0 |

* @param quarz - Quarzfrequenz 12MHz oder 24MHz
*****/
void init_UART(char quarz){

// Schnittstelle Initialisierung
SCON |= 0x40;    // SM1 (2^6) = 1;
//-----
// Timer2 Initialisierung
T2CON |= 0x30;    // TCLK (2^4) = 1; RCLK (2^5) = 1;
TR2 = ON;        // Timmer2 run T2CON 2^2
RCAP2H = 0xFF;

if(quarz == 12)    RCAP2L = 0xD9;    //Reloadwert bei 12MHz Quarz
else                RCAP2L = 0xB2;    //Reloadwert bei 24MHz Quarz
}

```

4. LÖSUNG AUFGABE 2

```

/*****
  Datei:      Aufgabe2_AB-001-U-01_ACII_ZeichenHyperterminal.c

  Datum:      Sep.2012 - first release
  Version     1.0

  Autor:      Viktor Schabelski info@i2c-bus.de
  Lizenz:     Creative Commons Lizenz
              Namensnennung - Keine kommerzielle Nutzung - Keine Bearbeitungen
              www.i2c-bus.de/cc

  Übungsaufgabe
  Aufgabe 2
  Erweitern Sie die Aufgabe 1.
  Folgende Punkte muessen beachtet werden:
  1. Zu den ASCII Zeichen soll Dezimal-Wert ausgegeben werden.
  2. Die Hyperterminal -Ausgabe soll wie auf dem Bild angepasst werden.

  Fuer diese Aufgabe werden zwei Boards benoetigt.
  1. Mikrocontrollerboard z.B. SB-001/SB-002
  2. AB-001/AB-002 Serielle Schnittstelle
*****/

/*****
  Includes
*****/
#include <at89c5131.h>
#include <stdio.h>

/*****
  Globale Variablen
*****/
//Quarz
#define _24MHZ      24
#define _12MHZ     12

#define ON 1

/*****
  Funktionsprototypen
*****/
void init_UART(char quarz);
void sendeZeichen_UART(char myDaten);
void sendeDezimal_UART(char myDaten);

/*****
  Funktionen
*****/
void main (void){
  char myZeichen = 0x21;    // ab Zeichen"!\"
  char i = 1;

  init_UART(_24MHZ);

```

```

do{
// Dezimal Zahl berechnen und ausgeben
  sendeDezimal_UART(myZeichen);
  sendeZeichen_UART(, );

// ASCII Zeichen ausgeben
  sendeZeichen_UART(myZeichen);
  sendeZeichen_UART(0x09);          // Horizontaler Tabulator

  myZeichen++;          // Zeichen um 1 erhoen

// Neue Zeile
  if(!(i % 5)){
    sendeZeichen_UART(0x0A); // Zeilenvorschub
    sendeZeichen_UART(0x0D); // Carriage Return
  }
  i++;

}while(myZeichen <= 126); // bis Zeichen „~“
while(1);
}

/*****
  Initialisierung RS232/V24 Schnittstelle.

  Initialisierung der seriellen Schnittstelle in Mode 1
  Schnittstellenparameter: 9600Baud, 8 Datenbit, 1 Stopp-Bit,
  asynchroner Betrieb ohne Empfang,
  Quarz Frequenz 12/24 MHz
  Timmer2
  SCON = 0x40h, Serial Control Register (0x98h)
  FE/SM0 - 0 |
  SM1 - 1 |
  SM2 - 0 |
  REN - 0 |
  TB8 - 0 |
  RB8 - 0 |
  TI - 0 |
  RI - 0 |

  T2CON = 0x30h, Timer 2 Control Register (0xC8h)
  TF2 - 0 |
  EXF2 - 0 |
  RCKL - 1 |
  TCKL - 1 |
  EXEN2 - 0 |
  TR2 - 0 |
  C/T2# - 0 |
  CP/RL2# - 0 |

  * @param quarz - Quarzfrequenz 12MHz oder 24MHz
  *****/
void init_UART(char quarz){

// Schnittstelle Initialisierung
  SCON |= 0x40;          // SM1 (2^6) = 1;

//-----
// Timer2 Initialisierung
  T2CON |= 0x30;          // TCLK (2^4) = 1; RCLK (2^5) = 1;
  TR2 = ON;              // Timmer2 run T2CON 2^2
  RCAP2H = 0xFF;

```

```

    if(quarz == 12)    RCAP2L = 0xD9;        //Reloadwert bei 12MHz Quarz
    else              RCAP2L = 0xB2;        //Reloadwert bei 24MHz Quarz
}

/*****
  Zeichen senden ueber SBUF
*****/
void sendeZeichen_UART(char myDaten){

    SBUF = myDaten;    // Zeichen in SBUF Register laden
    while(TI == 0);    //Warten bis ganzer Byte ,bertragen ist
    TI = 0;            // TI (Transmit Interrupt) Flag im Register SCON loeschen
}

/*****
Umrechnung:
aus ASCII Zeichen Zahlen erzeugen. ASCII Tabelle:

Char Dezimal   Hex
0    48        0x30
1    49        0x31
2    50        0x32
3    51        0x33
4    52        0x34
5    53        0x35
6    54        0x36
7    55        0x37
8    56        0x38
9    57        0x39
*****/
void sendeDezimal_UART(char myDaten){
    sendeZeichen_UART(((myDaten/100) % 10) + 48); // Hunderter
    sendeZeichen_UART(((myDaten/10) % 10) + 48); // Zehner
    sendeZeichen_UART((myDaten % 10) + 48);      // Einer
}

```

Haben Sie einen Fehler entdeckt?

Wir sind dankbar für Ihren Hinweis.
Schicken Sie uns bitte diesen Hinweis einfach per
E-Mail: info@i2c-bus.de.

Vielen Dank!