



Dokument NR.: AB-001-U-03

IN/OUT - UART



Einfache Ein- und Ausgabe über
serielle Schnittstelle
Interrupt-Betrieb



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

INHALTSVERZEICHNIS

- 1. Aufgabe 1 3
 - 1.1 Verwendete Module: 3
 - 1.2 „Target Mission“ 3
- 2. Lösung Aufgabe 1 4

Modification History:

Version	Date	Comments
1.0	12.2012	first release

1. AUFGABE 1

Schreiben Sie ein Programm, das einen Benutzer auffordert einen Zeichen mittels Tastatur-Taste einzugeben. Nach dem dieses Zeichen vom Mikrocontroller empfangen wurde, muss der Mikrocontroller das erkannte Zeichen auf dem HypertTerminal wiedergeben.

- Meiden Sie vorgefertigte Funktionen wie printf()
- Realisieren Sie Empfang- und Sende-Funktionen mittels „Interrupt“

1.1 VERWENDETE MODULE:

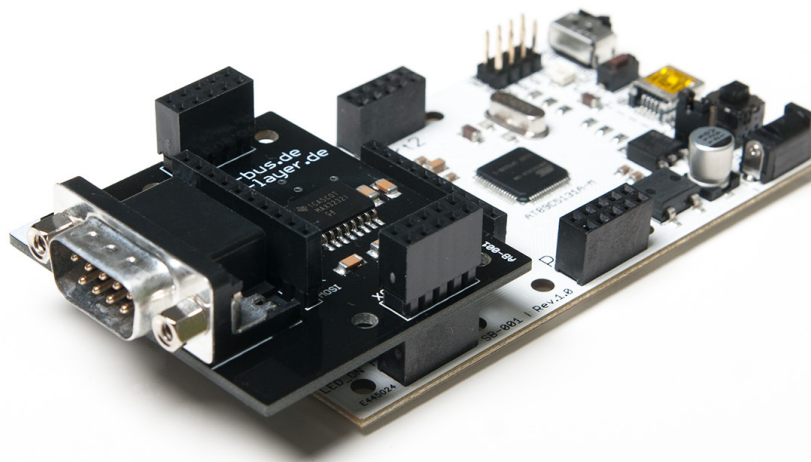
Mikrocontroller Board (SB-001/SB-002) und RS232 TTL Umsetzer (AB-001).

1.2 „TARGET MISSION“

Ein Video sagt mehr, als tausend geschriebene Worte.

In diesem Video wird gezeigt was genau in der Aufgabenstellung 1 (AB-001-U-03) gemeint ist.

[„Target Mission“ auf YouTube](#)



2. LÖSUNG AUFGABE 1

```

/*****
Datei:      Aufgabel_AB-001-U-03.c
Produkt:   AB-001-U-03
Projekt:   Aufgabe 1 IN/OUT with SBUF and Interrupt
Datum:    12.2012 - first release
Version   1.0
Compiler:  Keil V9.05

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

```

Schreiben Sie ein Programm, das einen Benutzer auffordert einen Zeichen mittels Tastatur-Taste einzugeben.

Nach dem dieses Zeichen vom Mikrocontroller empfangen wurde, muss der Mikrocontroller das erkannte Zeichen auf den HyperTerminal wiedergeben.

- Meiden Sie vorgefertigte Funktionen wie printf()
- Realisieren Sie Empfangs und Sendefunktionen mittels Interrupt

```

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

#define QUARZ_FRQ 24      /* Quarz Frequenz */
#define SIZE 1          /* Empfangsbuffer Groesse */

#define BUSY 1
#define FREE 0

#define NO 0
#define YES 1

unsigned char auchBuffer[SIZE] = {0};      /* Empfangsbuffer */
bit bUart = FREE;                          /* UART Status */
bit bNewChar = NO;

void init_UART(unsigned char uchQuarz);
void init_ISR(void);
void sendString_UART(unsigned char* pString);
void sendChar_UART(unsigned char uchCharacter);
void isr_UART(void);

```

```

/*****
Funktionen
*****/

/*=====
* Name      : main
* Input     : none
* Output    : none
* Descr.    : Main Funktion
-----*/
void main (void) {
    init_UART(QUARZ_FRQ);
    init_ISR();
    sendString_UART („Druecken Sie eine Taste: „);
    while(1) {
        if(bNewChar) {
            sendChar_UART(auchBuffer[0]);
            bNewChar = NO;
        }
    }
}

/*=====
* Name      : init_UART
* Input     : uchQuarz - Quarzfrequenz - erlaubte Werte 12 fuer 12MHz und 24 fuer 24MHz
* Output    : none
* Descr.    : Initialisierung RS232/V24 Schnittstelle.
*           : Initialisierung der seriellen Schnittstelle in Mode 1
*           : Schnittstellenparameter: 9600Baud, 8 Datenbit, 1 Stopp-Bit asynchroner Betrieb
*           : mit Empfang
*           : Quarz Frequenz 12/24 MHz
-----*/
void init_UART(unsigned char uchQuarz) {
    /* Schnittstelle Initialisierung */
    SCON |= 0x50;          /* SM1 (2^6) = 1; REN (2^4) = 1; */

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

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

    TI = 1;
}

/*=====
* Name      : init_ISR
* Input     : none
* Output    : none
* Descr.    : Interrupt Service Rutine Initialisierung
*           : Im Register IEN0 werden die Bits EA(2^7) und ES(2^4) gesetzt
-----*/
void init_ISR (void) {
    IEN0 |= 0x90;
    IPLO |= 0x10;      /* Interrupt Priorisierung */
    IPHO |= 0x00;
}

```

```

/*=====
* Name      :  sendString_UART
* Input     :  pString - Pointer auf String
* Output    :  none
* Descr.    :  Sende String ueber RS232/V24 Schnittstelle.
-----*/
void sendString_UART(unsigned char* pString){
    while(*pString != 0){
        sendChar_UART(*pString); /* sende Zeichen */
        pString++;
    }
}

/*=====
* Name      :  sendChar_UART
* Input     :  uchCharacter - Zeichen zum Senden
* Output    :  none
* Descr.    :  Sende Zeichen ueber RS232/V24 Schnittstelle.
-----*/
void sendChar_UART(unsigned char uchCharacter){
    while(bUart == BUSY);
    SBUF = uchCharacter;      /* sende Zeichen */
    bUart = BUSY;            /* UART besetzt */
}

/*=====
* Name      :  isr_UART
* Input     :  none
* Output    :  empfangener Zeichen in array auchBuffer
* Descr.    :  ISR
-----*/
void isr_UART (void) interrupt 4{
    static unsigned char uchCount = 0x00;

    /* Zeichen empfangen */
    if (RI){
        auchBuffer[uchCount] = SBUF;
        RI = 0x00;
        bNewChar = YES;
        if (uchCount >= SIZE -1){
            uchCount = 0x00;
        }else{
            uchCount++;
        }
    }

    /* Zeichen senden */
    if (TI) {
        TI = 0x00;
        bUart = FREE;
    }
}

```

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!