

# Mikroprozessorsysteme Praktikum 5

## Aufgabe 1

**Was passiert, wenn Sie nach CR und LF noch weitere Zeichen auf die gegebene Weise ausgeben?**

Die Zeichen werden in den Holding Register gespeichert und wenn zu viele auf einmal gespeichert werden, kommt das Senderegister nicht mit. Das kann man lösen indem man eine Warteschleife einbaut, bis das nächste Zeichen eingelesen wird.

**Beschreiben Sie den Unterschied der Funktionen `inits()` und `init_ser()`.**

`init_ser()` initialisiert die serielle Schnittstelle des USART0 auf dem Board (Prozessor ist im User-Mode)

`inits()`: Ruft den Switch-Case 0x100 auf, was dafür sorgt, dass die `init_ser()`-Funktion aufgerufen wird (Prozessor ist im Supervisor-Mode)

`Inits` ruft `init_ser()` im Supervisor Mode auf

## Aufgabe 2

**Woran erkennen Sie, in welchem Mode sich der Prozessor befindet?**

Den Zustand des aktuellen Modus kann man aus dem CPSR-Register herauslesen.

**An welcher Stelle wird der Superuser Mode verlassen?**

Wenn `swi.c` verlassen wird

## Aufgabe 3

```
@ Funktion
.text
.align 2
.global puts
.type puts,function

puts:
    stmfd sp!,{lr} @ Retten der Register

// Hier muß Ihr Code eingefügt werden.

    mov r4, r0      @Speichere Adresse in r4

loopPuts:
    ldrb r5, [r4],#1
    cmp r5, #0xa
    bne endNewLine
    mov r0, #0xd
    bl putc

endNewLine:
    mov r0, r5
    cmp r0, #0
    beq endPuts
    bl putc
    b loopPuts

endPuts:
    ldmfd sp!, {pc}

@ Funktion
.text
.align 2
.global putc
.type putc,function

putc:
    mov r1, r0      @ Zeichen nach r1
    ldr r0, =Zeichen @ Zeiger holen
    str r1, [r0]     @ Zeichen unter Zeiger ablegen
    mov r6, r1
    swi 0x200        @
    ldr r1, =Zeichen @ Zeiger holen
    ldr r0, [r1]     @ Zeichen aus Zeiger holen
    cmp r0, #0       @ UPDATE: Falls der Sender beschaeftigt ist, kommt eine Null
    mov r0, r6        @ UPDATE: Speicher das Zeichen zurueck, damit es wieder vhd. ist falls der Sender beschaeftigt war
    beq putc          @ UPDATE: Falls Senden fehlgeschlagen; try again
    bx lr
```

siehe Code

## Aufgabe 4

```
void intToString (int var)
{
    long int a = 10;
    int b = 0;           //b legt sp??ter fest and welche Stelle des arrays der char geschrieben wird
    char carray[22];      //20 Stellen fue eine 64bit Zahl + eine Stelle fuer '-' und eine fuer '\0'

    if(var < 0){           //pr??fen ob var kleiner 0, wenn ja, ein minus in das array schreiben und var mit minus 1 multiplizieren
        var = var * -1;
        carray[0]='-';
        b=b+1;
    }

    while((var % a) != var){ //a so oft mit 10 multiplizieren, bis var%a = 0
        a=a*10;             //sp??ter wird var durch a geteilt um die erste Stelle rauszukriegen
    }
    a=a/10;                //am Ende der Schleife is a um eine Gr???enordnung gr???er als var, deswegen /10

    while(a>0){            // var stellenweise in das array schreiben
        carray[b]=((var/a)+48); // Die Ziffern 0-9 fangen bei 48 im ASCII-Code an var/a = ergibt die jeweils n??chste Stelle der Zahl
        var = var - ((var/a)*a); // Entfernt die herausgeschriebene Ziffer, z.B. 253/100 = 2, 2*100 = 200, 253-200 = 53
        a = a/10;          // a um eine Gr???enordnung verringern
        b = b+1;           // n??chste Stelle des arrays
    }
    carray[b]='\0';        //array "abschlie??en"
    puts(carray);         //array ausgeben
}
```

siehe Code

**Wie wird die größte darstellbare negative Zahl 0x80000000 ausgegeben?**

-2147483648