

## Übungsblatt 3

## Aufgabe 1: Bit-Operationen und Input-Handling

In dieser Übung soll eine Textdatei über einen Drucker ausgegeben werden. Bevor der Drucker funktionsbereit ist muss dieser konfiguriert werden. Hierfür besitzt der Drucker 3 Register in denen entsprechende Einstellungen Bit-Kodiert werden.

Lesen Sie die folgenden Werte mittels cin und/oder scanf jeweils einzeln ein und speichern Sie diese in den Registern an der korrekten Stelle.

```
unsigned short Output_Timing;  
unsigned char LineFeed_Handling;  
unsigned char Space_Handling;  
unsigned char Page_Size;  
unsigned char Junk_Size;  
unsigned char Char_Size;  
unsigned char End_Page;  
unsigned char Start_Page;
```

Die auszugebende Datei (C-Standard.txt) sowie den Code des Druckers (Printer.h) finden Sie auf GRIPS.

Folgende Tabellen beschreiben die Register-Layouts:

Register 1:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_OUT											LF			SP	

Field	Function	
15-5 T_OUT	11-Bit value to set the delay between write operations.  Output delay = $10.000 / (T\_OUT + 1)$ ms	
4-2 LF	Bit-Field for line-feed handling:	
	0bXX0	No Line-Feed after each junk of output data
	0bXX1	Additional Line-Feed after each junk of output data
	0bX0X	Ignore Line-Feed
	0bX1X	Print Line-Feed
	0b0XX	Print Line-Feed as Space
	0b1XX	Print Line-Feed
1 SP	Bit for Space handling:	
	0	Ignore all Space character
	1	Print Space character

### Register 2:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAGE_SIZE								JUNK_SIZE						CHAR_SIZE	

Field	Function
15-10 PAGE_SIZE	6-Bit value to determine the Page size. PAGE_SIZE defines the number of lines per page  PAGE_SIZE must be > 0
7-4 JUNK_SIZE	4-Bit value to determine the Junk size. JUNK_SIZE defines the number of character that are printed in one step. After each Junk the Printer will wait 10.000/(T_OUT+1) ms before printing Junk.  JUNK_SIZE must be > 0
1-0 CHAR_SIZE	2-Bit value to determine size of Character in Byte: Valid values: 1: char (1 Byte) 2: wchar (2 Byte)  must be 0 < CHAR_SIZE < 3

### Register 3:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
END_PG								START_PG							

Field	Function
15-8 END_PG	8-Bit value to specify the last page that shall be printed.  Must be >= START_PG
7-0 START_PG	8-Bit value to specify the first page that shall be printed. First page equals START_PG = 0  Must be <= END_PG

Bitte beachten Sie:

- Fügen Sie die Printer.h Datei zu Ihrem Projekt hinzu
- Inkludieren Sie Printer.h in Ihrem main-File **vor allen anderen includes**
- Geben Sie für startPrinter einen Pfad zur Datei C-Standard.txt an. Dieser kann absolut oder relativ zur exe-Datei angegeben werden.
  - Beachten Sie das Backslash in C als default-Token gilt und deshalb „Escaped“ werden muss. Wie im Beispiel-Code verwenden Sie doppelten Backslash \\
- Die Deklaration für register1 – register3 (Beispiel-Code Zeile 1-3) sind optional und dienen nur der Veranschaulichung. Diese Variablen bereits existieren und nicht von Ihnen Definiert werden müssen.

Starten Sie mit folgendem Code:

```
extern unsigned short register1;
extern unsigned short register2;
extern unsigned short register3;

void main()
{
    unsigned short Output_Timing;
    unsigned char LineFeed_Handling;
    unsigned char Space_Handling;
    unsigned char Page_Size;
    unsigned char Junk_Size;
    unsigned char Char_Size;
    unsigned char End_Page;
    unsigned char Start_Page;

    /*
     * Hier input handling einfügen
     */

    /*
     * Register1 setzen
     * Register2 setzen
     * Register3 setzen
     */

    // Pfad anpassen! '\\' muss escaped werden '\\\'
    startPrinter("C:\\C-Standard.txt");

    return;
}
```