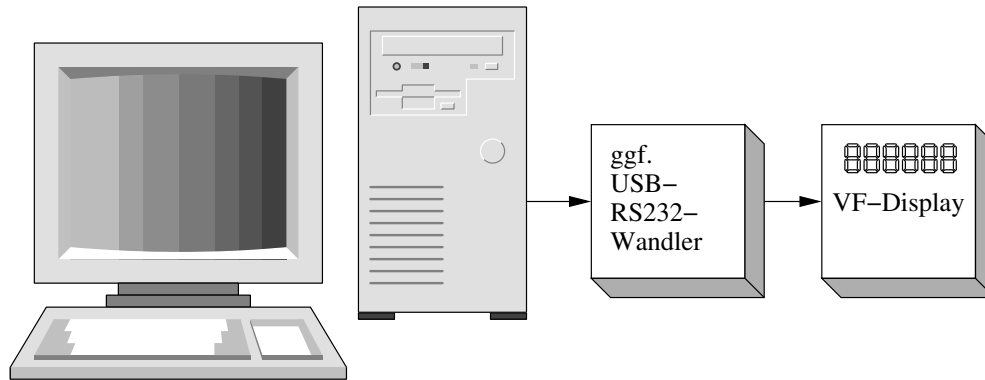


7.12.A Projekte/VFD-Bibliothek – Arbeitsblatt

1 Thema

Es soll in C eine Funktionsbibliothek erstellt werden, mit deren Hilfe ein Vakuum-Fluoreszenz-Display (VFD) über eine PC-Schnittstelle angesteuert werden kann.



Die Funktionsbibliothek soll so modular sein, dass die gewählte Schnittstelle (seriell, USB per Adapter oder parallel) ohne große Änderung ausgetauscht werden kann.

Ebenso soll es möglich sein, die Funktionsbibliothek auf einen Mikrocontroller zu portieren.

Mit einem kleinen Hilfsprogramm soll die Funktionsbibliothek getestet und vorgeführt werden.

2 Software: API

Die Funktionsbibliothek soll folgende Funktionen zur Verfügung stellen:

```

int  vfd_init(void);
void vfd_exit(void);
int  vfd_clear(void);

int  vfd_printstr(const char *s);
int  vfd_printlu(unsigned long val, int fmt);
int  vfd_printbits(unsigned char c);
int  vfd_gotolc(int line, int column);

```

Mit `vfd_init()` wird die Bibliothek initialisiert. Mit `vfd_exit()` wird die Benutzung der Bibliothek abgeschlossen. Die Funktion `vfd_clear()` löscht das Display. Mit `vfd_printstr(s)` wird die Zeichenkette `s` auf dem Display ausgegeben. Die Ausgabe von Umlauten soll möglich sein. Mit `vfd_printlu(val, fmt)` wird die Zahl `val` ausgegeben. Der Parameter `fmt` beeinflusst das Format der Zahlenausgabe, dargestellt ist das in Tabelle 1 am Beispiel der Ausgabe der Zahl 1234567. Die Funktion `vfd_printbits(c)` gibt das Byte `c` als Binärzahl aus, und die Funkti-

fmt	Ausgabe
0	1234567
1	1 234 567
2	1.234.567
3	1M234k567

Tabelle 1: Bedeutung des Parameters `fmt` bei der Funktion `vfd_printlu()`

on `vfd_gotolc(line, column)` bewegt die Cursorposition an die Spalte `column` in der Zeile `line`.

3 Software: Modularität

Die Software soll aus mindestens zwei Schichten bestehen. Die obere Schicht soll die API zur Verfügung stellen. Die untere Schicht soll mit der Hardware kommunizieren. Die obere Schicht soll die Hardware niemals direkt ansprechen, sondern nur über die untere Schicht. Das Testprogramm soll ein eigenes Modul sein, das (außer evtl. zu Testzwecken) nur per API mit der oberen Schicht kommuniziert und mit der unteren Schicht gar nicht.

4 Software: Portabilität

Mit Hilfe geeigneter Techniken (z.B. symbolische Konstanten, bedingte Compilierung) soll erreicht werden, dass das Verhalten der unteren hardwarenahen Schicht einfach geändert werden kann, wenn eine verwendet wird.

5 Dokumentation

Die Dokumentation der Software soll folgende Punkte enthalten:

- eine allgemeine Beschreibung der Software
- zu jeder Funktion (hier: im Quelltext!) eine kurze Aufstellung gemäß der vorgestellten Vorlage
- Aufrufgraph (*call graph*), d.h. ein Diagramm zur Aufrufhierarchie der Funktionen, evtl. mit Zuordnung der Funktionen zu den Modulen