

7.12 Projekte/VFD-Bibliothek

7.12.1 Informationen zur Software

Möglichkeiten der Programmierung Peripheriegeräte kann man generell auf zwei Arten ansprechen:

- Direktzugriff über Portnummern und Ein-/Ausgabebefehle
- Zugriff per Treiber

Der Zugriff per Treiber ist manchmal wesentlich einfacher. In jedem Fall bietet er mehr Portabilität, d.h., dass die Software bei einer System-Änderung (neues Betriebssystem, neue Hardware) nicht verändert werden braucht.

Dafür bietet die direkte Programmierung oft mehr Möglichkeiten. Das liegt daran, dass die (abstrakte) Treiber-Software eventuell nicht alle Möglichkeiten dieses konkreten Peripheriegerätes kennt oder ausnutzt.

Programmierung per Treiber: Arten Bei der Programmierung mittels Treiber muss man zwischen zwei Arten von Peripheriegeräten unterscheiden können:

- Block-orientierte Geräte wie Festplatten, DVD-Laufwerke, Dokumenten-Scanner
- Zeichenstrom-orientierte Geräte wie Drucker, Netzwerkgeräte, Maus

Ein Block-orientiertes Gerät steuert man, indem man Befehle an das Gerät ausgibt, mit denen man Datenblöcke gezielt ein- und ausgeben kann. Ein Zeichenstrom-orientiertes Ausgabegerät dagegen kann man behandeln wie eine Textkonsole, in die man Text ausgibt oder auch wie eine Datei, in die man schreibt. Ein Zeichenstrom-orientiertes Eingabegerät kann man behandeln wie eine Tastatur, von der man Daten liest oder eben auch wie eine Datei, aus der man liest. Die Schnittstellen RS232 (serielle Schnittstelle) und IEEE-1284 (parallele Schnittstelle, Centronics) werden von gängigen Betriebssystemen (Linux, MacOS, Unix, Windows) als Zeichenstrom-orientierte Geräte angesehen.

Programmierung per Treiber: Gerätedateien Die gängigen Betriebssysteme erlauben es, dass die Treiber, über die man Peripheriegeräte anspricht, im Dateisystem mit einem eigenen Namen ansprechbar sind. Im folgenden ein paar Beispiele:

Gerät	Linux-Name	Windows-Name
Festplattenpartition	/dev/sda1	C:
Textkonsole	/dev/tty1	CON:
Nirgendwo	/dev/null	NUL:
Drucker	/dev/lp0	LPT1:
RS-232-Schnittst.	/dev/ttyS0	COM1:, AUX:
USB-RS232-Wandler	/dev/ttyUSB0	COM5: (?)

Programmierung per Treiber: C Sobald man den Namen der Gerätedatei hat, kann man in C ein Zeichenstrom-orientiertes Peripheriegerät wie eine Datei ansprechen. Man kann diese Gerätedatei also öffnen, beliebig oft lesen oder schreiben und bei Bedarf wieder schließen — genau wie eine normale Datei.

7.12.2 Informationen zur Hardware

Informationen zum VF-Display vom Typ Siemens BA63 findet man unter:
<http://www.mikrocontroller.net/attachment/43623/ba63.pdf>
http://www.maltepoeggel.de/html/vfd/ba63_zeichensatz.pdf