

Vorwort



Abb. 1 Eine kleine Auswahl kleiner Module.

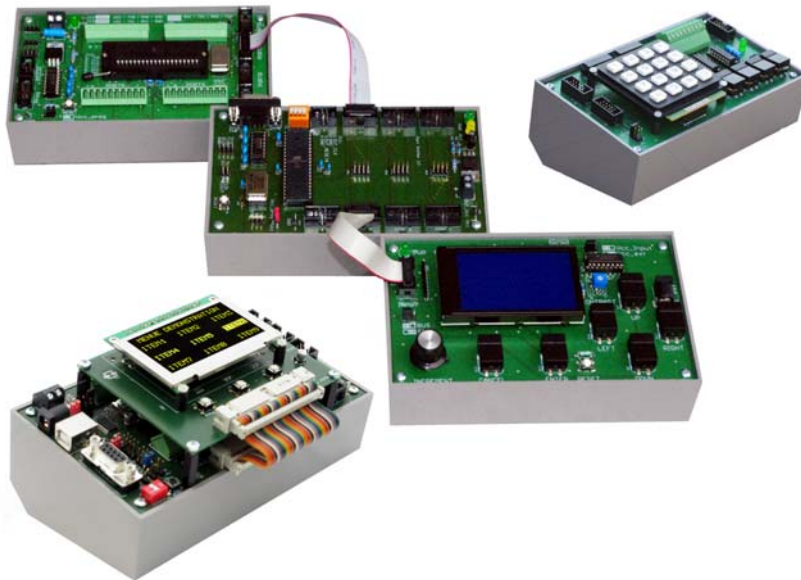


Abb. 2 Einige der Module, die im vorliegenden Buch beschrieben werden.



Abb. 3 Funktionseinheiten der industriellen Technik.

1. Grundlagen

1.1 Aufgaben der Projektentwicklung



Abb. 1.1 Die meisten Mikrocontrollerplattformen enthalten die Funktionseinheiten, die hier darstellt sind. Die Unterschiede liegen in den Einzelheiten.

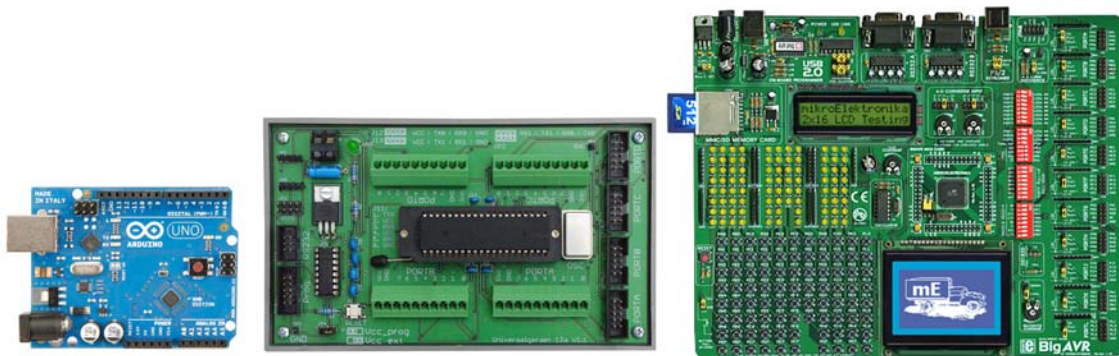


Abb. 1.2 Beispiele kleiner, mittelgroßer und großer Mikrocontrollermodule.



Abb. 1.3 Leistungsstufen zum Aufstecken. Sie müssen offensichtlich ganz oben sitzen. Weitere Aufsteckmöglichkeiten sind deshalb gar nicht vorgesehen.



Abb. 1.4 Man fragt sich, was sich die Erfinder wohl dabei gedacht haben... Die Leistungsstufe rechts muss ganz oben sein, allein schon wegen des Kühlkörpers. Aber was ist mit der Leistungsstufe links? Wenn man ein weiteres Modul aufsteckt, kommt man nicht an die Stiftleisten heran. Sinnig ist auch die LED-Matrix in der Mitte. Wenn man noch ein Modul darüberstapelt, wird von den LEDs wohl nicht mehr viel zu sehen sein...



Abb. 1.5 Aufsteckmodule mit Gehäusen, die dazu passen. So sieht es gut aus. Man kann aber nur das jeweilige Modul einsetzen. Konfigurationen mit mehreren Modulen werden offensichtlich nicht unterstützt.

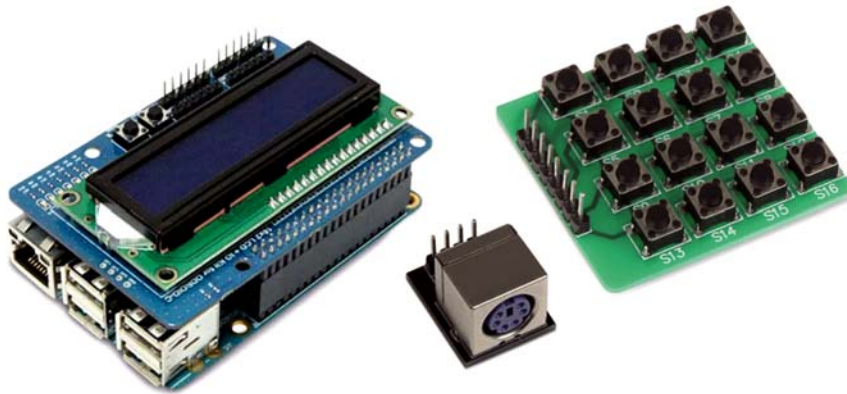


Abb. 1.6 Links ein Mikrocontrollermodul mit aufgestecktem LCD-Modul. Mehr geht offensichtlich nicht. Was aber tun, wenn die zwei Tasten nicht genügen? Nun gibt es fertige Lösungen, wie beispielsweise ein Tastenfeld oder eine Buchse für eine PS/2-Tastatur. Die hier gezeigten Teile muss man aber mit einzelnen Drähten anschließen und irgendwie in der Hand halten. Die Platinen haben nicht einmal Befestigungslöcher.

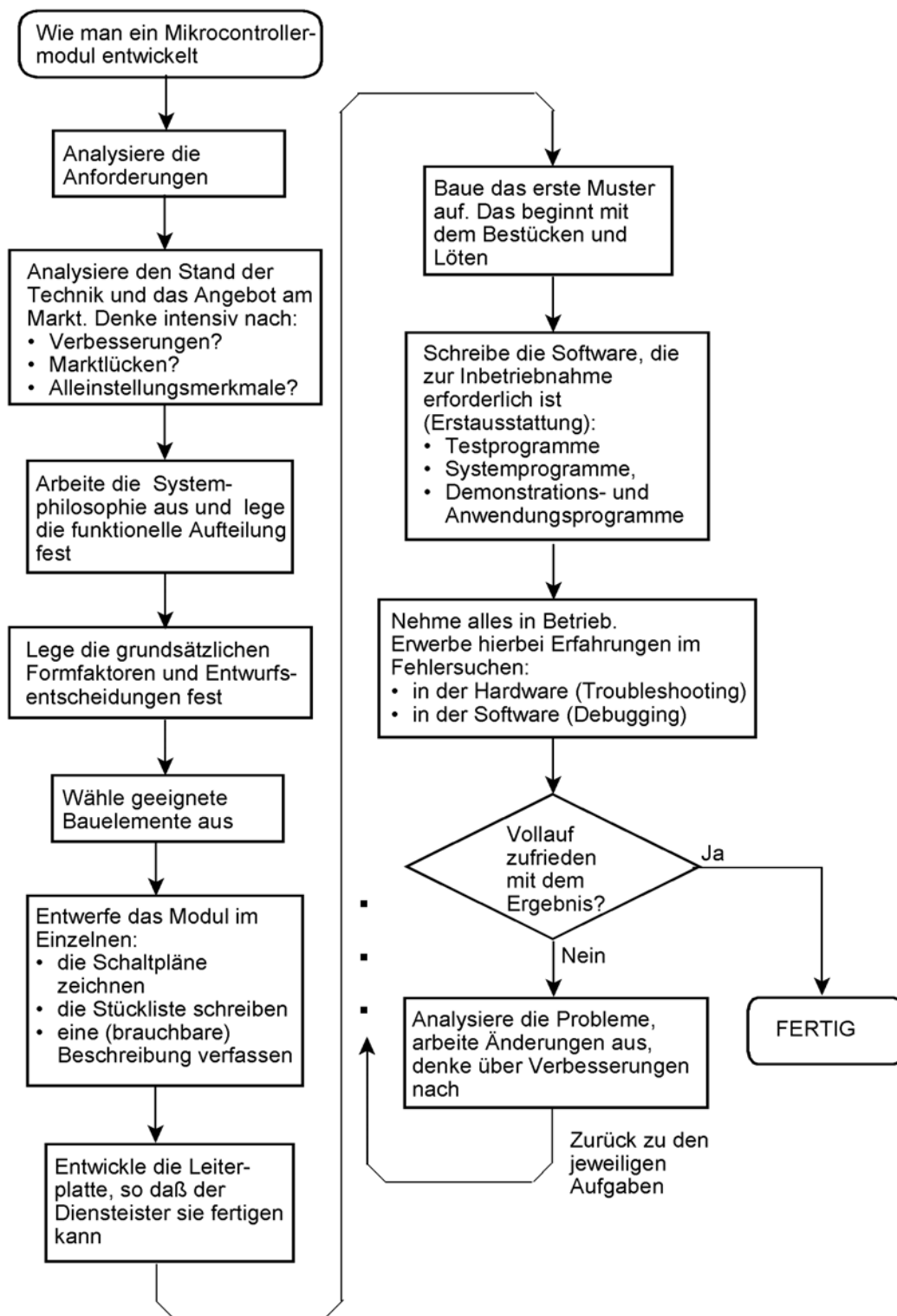


Abb. 1.7 Mikrocontrollermodule entwickeln. Ein Überblick über den Entwicklungsablauf.

1.2 Mikrocontrollersysteme

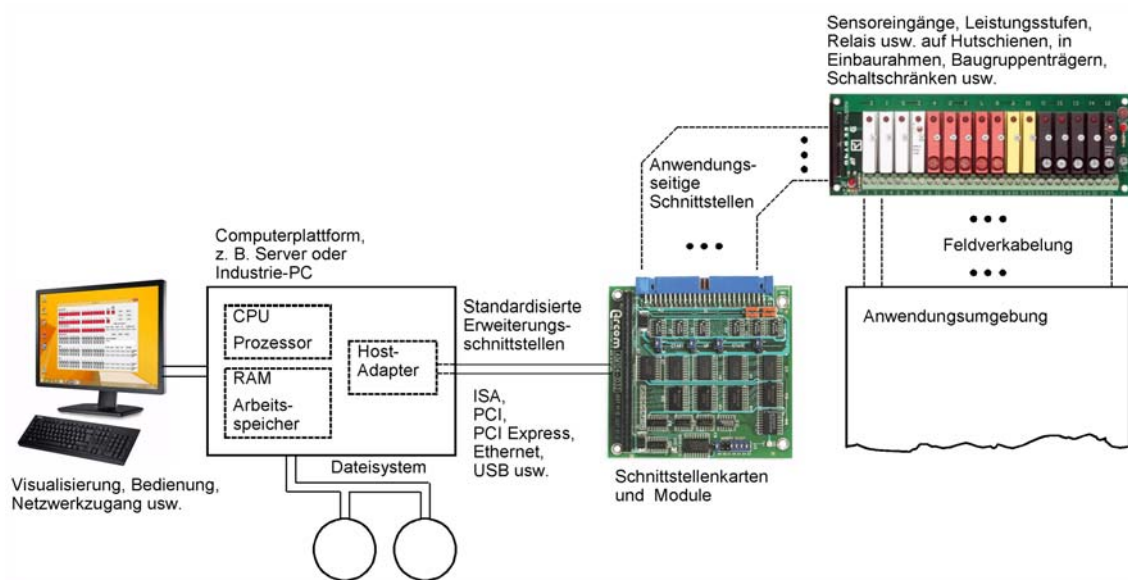


Abb. 1.8 Eine typische Anwendungslösung der industriellen Praxis. Embedded Systems auf Grundlage von Mikrocontrollern sind im Grunde genauso aufgebaut; sie sind nur kleiner und einfacher.

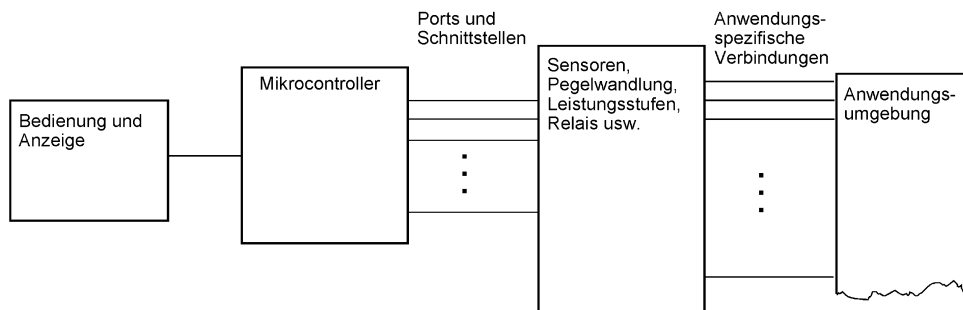


Abb. 1.9 Ein typisches Mikrocontrollersystem im allgemeinen Blockschaltbild.

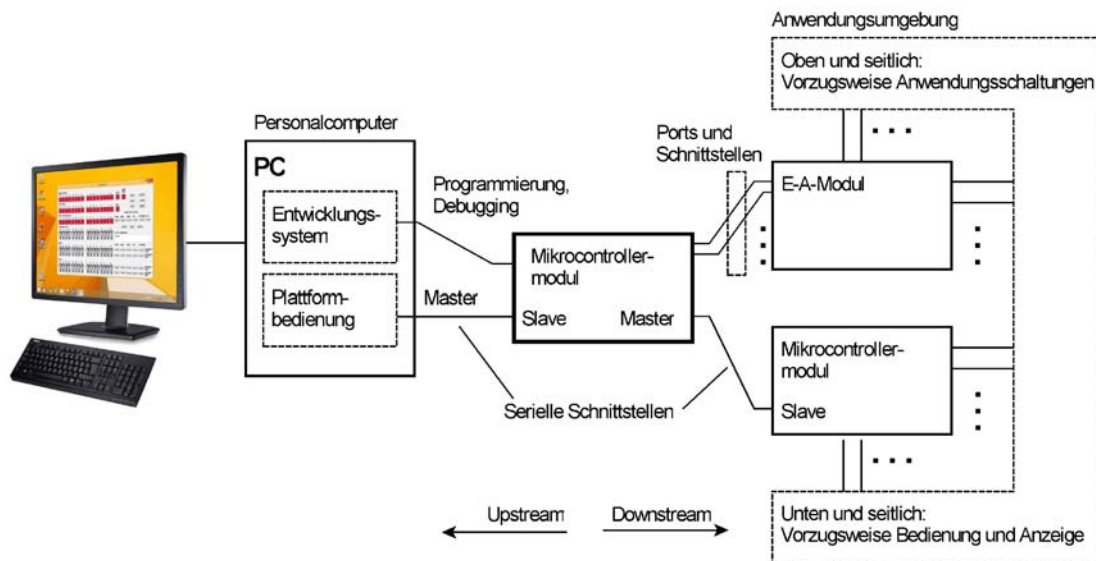
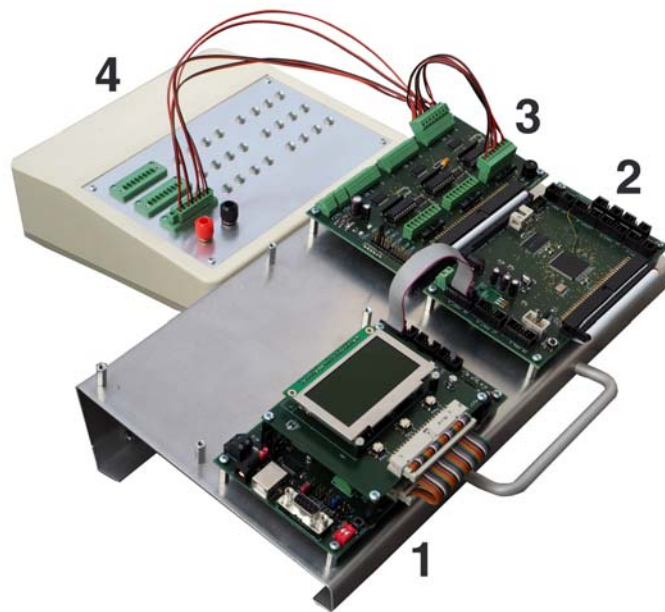


Abb. 1.10 Zum Experimentieren und Lernen werden die Module vorzugsweise so aufgestellt.

1.3 Die Anwendungsumgebung



Abb. 1.11 Der Lüftertrainer 14a. Zum Üben braucht man einen Gleichstrommotor, der nicht viel kostet und sich ohne besonderen Aufwand montieren lässt. Lüfter für PCs usw. erfüllen diese Anforderung. Schrauben, Muttern und Distanzhülsen genügen. Man muss ihn nicht in der Hand halten oder lose auf den Tisch legen. Über einen 8-Bit-Port können bis zu 8 Ausgänge angesteuert werden. Ausgangsstufe: ULN2803. Der Treibertransistor des Lüfters wird an einen dieser Ausgänge angeschlossen (Drahtbrücke).



- | | |
|------------------------|----------------------|
| 1 Mikrocontrollermodul | 3 24-V-Companion |
| 2 CPLD-Modul | 4 24-V-Leuchtanzeige |

Abb. 1.12 Hier wird die Anwendungsperipherie auf einfache Weise nachgebildet, um die grundsätzliche Funktionsfähigkeit zu prüfen. Der 24-V-Companion 3 ist ein Pegelwandlermodul mit 24 Ausgängen und 24 Eingängen, aufgeteilt auf jeweils drei 8-Bit-Ports. Die Anzeigeeinheit 4 ist ein selbst gebautes Prüfhilfsgerät. Die Ausgänge sind mit der Anzeigeeinheit 4 verbunden und zudem auf die Eingänge zurückgeführt. Das ist hier anhand eines Ports dargestellt.

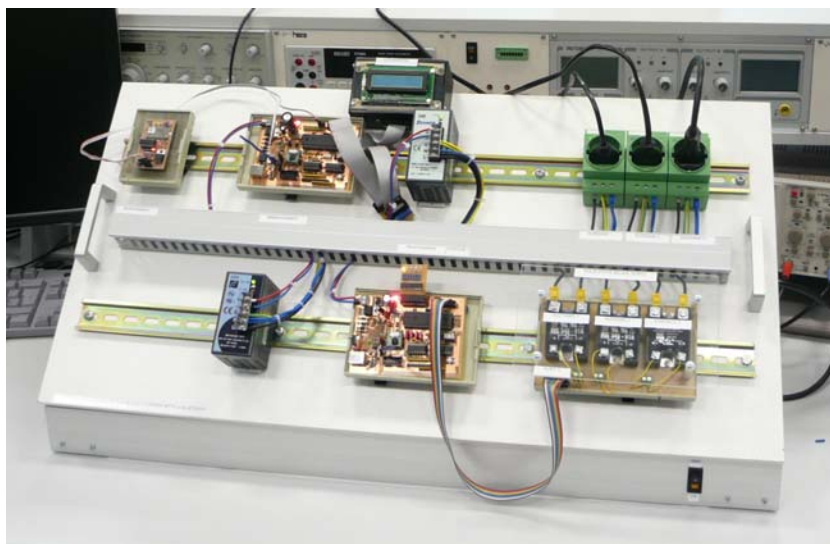


Abb. 1.13 Eine Prüftafel zum Experimentieren mit 230 V. Alles, was Netzspannung führt, ist berührungssicher abgedeckt. An die Steckdosen rechts oben sind echte Verbraucher angeschlossen.

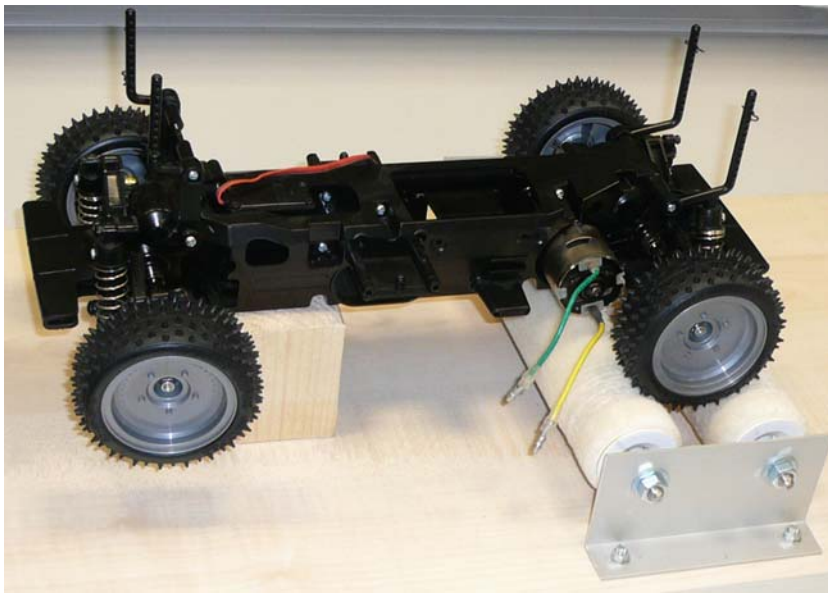


Abb. 1.14 Radfahrzeugmodelle brauchen einen Rollenprüfstand.

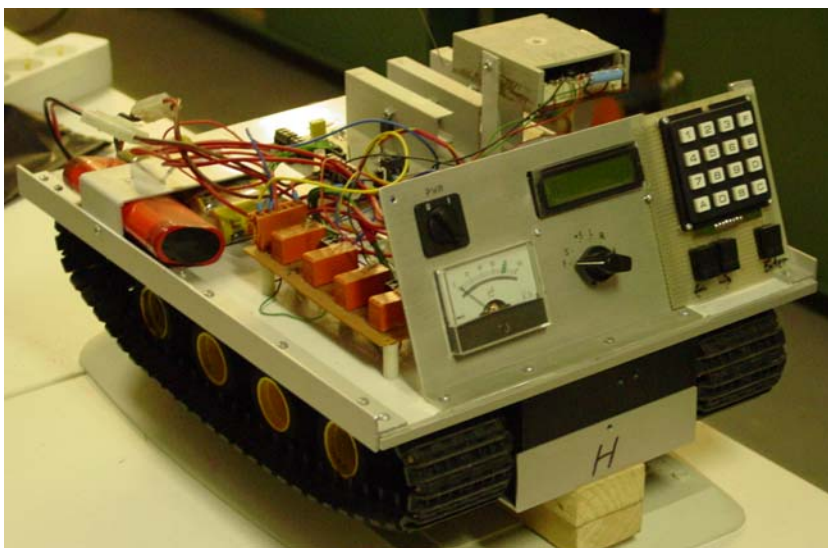


Abb. 1.15 Beim Kettenfahrzeugmodell reicht es üblicherweise aus, wenn die Ketten frei umlaufen können.

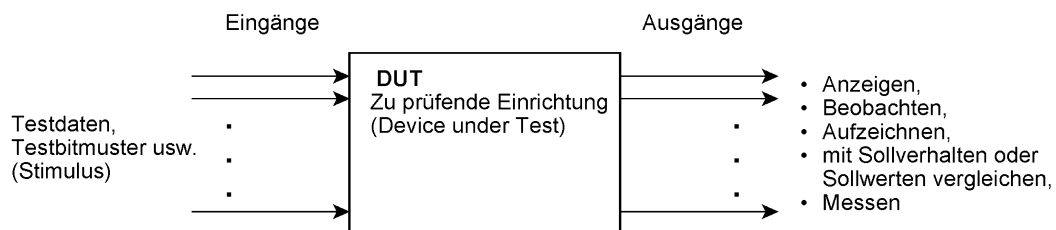


Abb. 1.16 Das Prinzip des Prüfens und Testens.

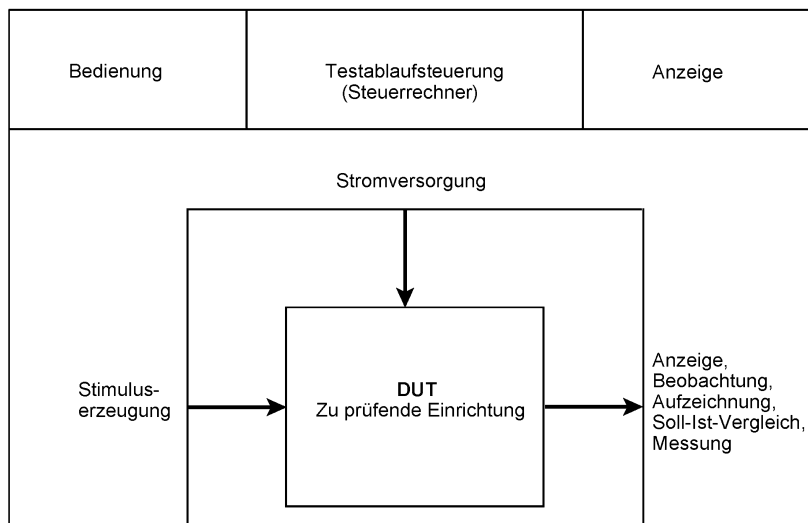


Abb. 1.17 Die grundsätzlichen Komponenten einer Testumgebung oder Prüfplattform.



- | | |
|--|--|
| 1 Mikrocontrollermodul als Anwendungs- oder Testmaschine | 3 Das zu prüfende E-A- Modul |
| 2 LCD-Modul als Protokollgerät (Prüfschrittanzeige usw.) | 4 Mikrocontrollermodul als Peripherienachbildung |

Abb. 1.18 Ein Mikrocontrollermodul als Peripherienachbildung oder Testmaschine. Es soll ausprobiert werden, ob fertig gekaufte E-A-Module mit unseren Mikrocontroller-Modulen betrieben werden können.

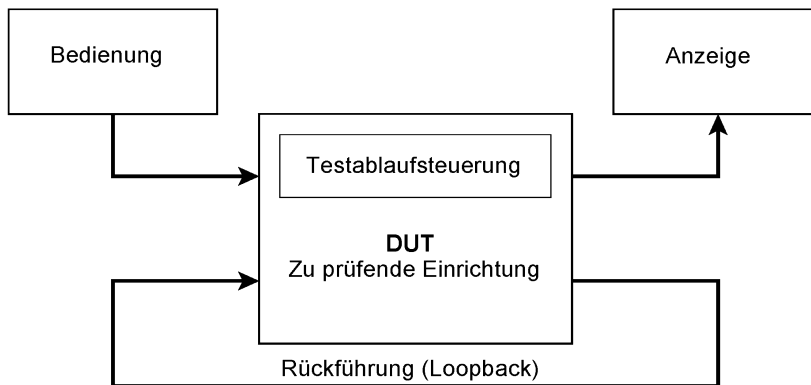


Abb. 1.19 Das Prinzip der Selbstprüfung. Ausgänge sind auf Eingänge zurückgeführt.



Abb. 1.20 Selbstprüfung mit einem angeschlossenen Computer.



Abb. 1.21 Siebensegmentanzeigen zum Anschließen an 8-Bit-Ports. Sieben Segmente + Dezimalpunkt oder Tastenabfrage (über Jumper wählbar).



Abb. 1.22 Bedien- und Anzeigemodule zum Anschließen an Mikrocontrollermodule.



Abb. 1.23 Ein Mikrocontroller-Modul mit Aufsteckplatine.



Abb. 1.24 Hier ist alles beieinander: Mikrocontroller, Bedienung, Anzeige, Schnittstellen, universelle Ein- und Ausgabe. Zwei All-in-One-Geräte.



Abb. 1.25 Manchmal ist das Einfachste das Beste. Kippschalter und LEDs. Ganz so altmodisch wie vor 40 Jahren (links außen) muss es aber nicht mehr zugehen. Die Kippschaltergeräte unseres Modulbaukastens (Mitte und rechts) haben LCD-Anzeigen und Computerschnittstellen. Man kann mehrere Geräte im Verbund einsetzen und die Schalter und Anzeigen beschriften. Aufkleben von Zetteln usw. unnötig...



Abb. 1.26 Eine historische Bedientafel mit 108 Kippschaltern (International Computers Limited).

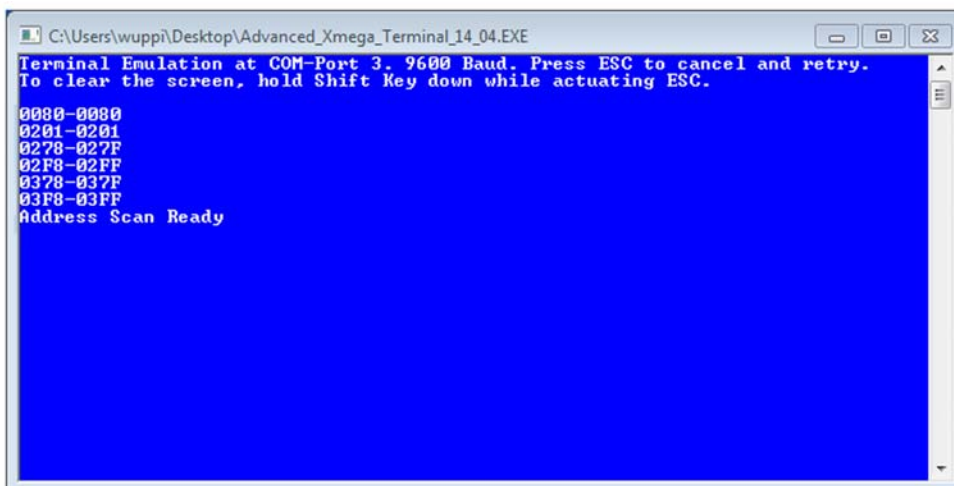
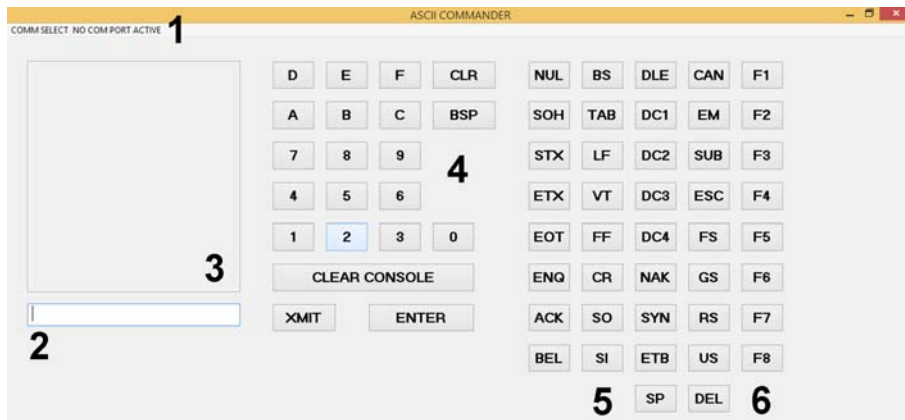


Abb. 1.27 Seit Jahrzehnten bewährt: das klassische Bedienterminal. Kommandos und Parameter werden eingetippt, Ausgaben des Mikrocontrollers angezeigt. Ein ganz einfaches Terminalprogramm genügt.



- 1 Auswahl der seriellen Schnittstelle
- 2 Eingabezeile
- 3 Ausgabefenster (wie Protokollschreibmaschine)
- 4 Eingabetasten
- 5 ASCII-Sonderzeichen
- 6 Funktionstasten (werden anwendungsspezifisch belegt)

Abb. 1.28 Dieses Dienstprogramm ermöglicht es, Kommandos zum Mikrocontroller zu schicken und die Antworten zu protokollieren. Es ist praktisch ein anwendungsspezifisches Terminalprogramm, das deutlich mehr Bedienkomfort bietet.

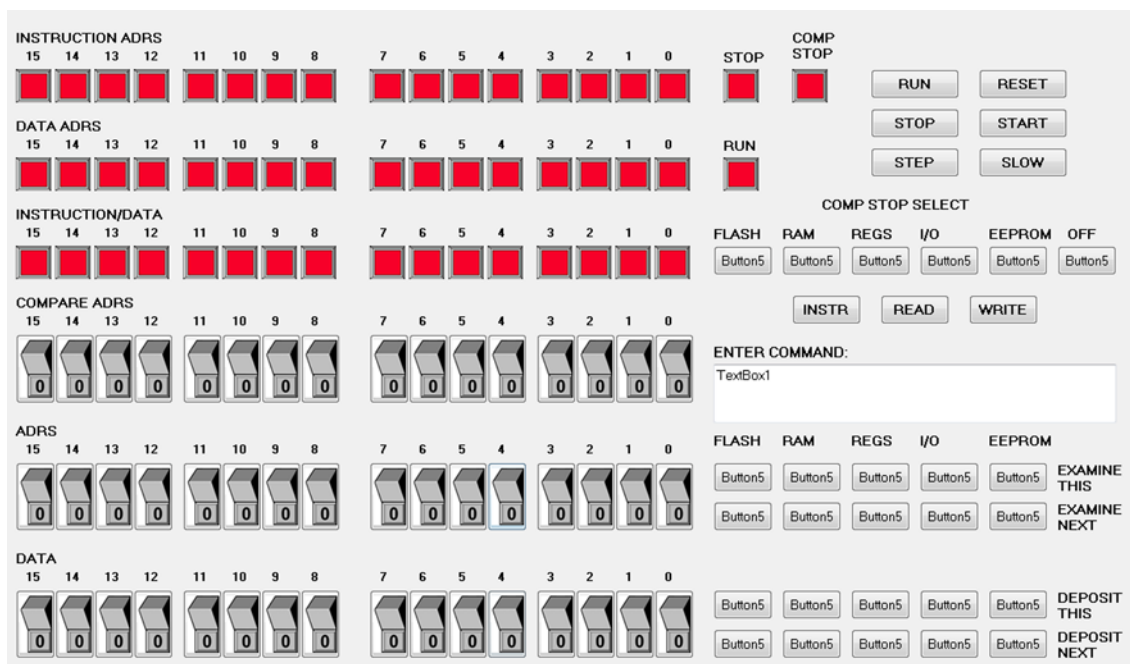


Abb. 1.29 Experimentelles Retro-Design in Anlehnung an historische Bedienfelder. Die Schalter, Tasten und Anzeigen wären viel teurer als ein kleiner PC, der dieses Bild anzeigen kann ...

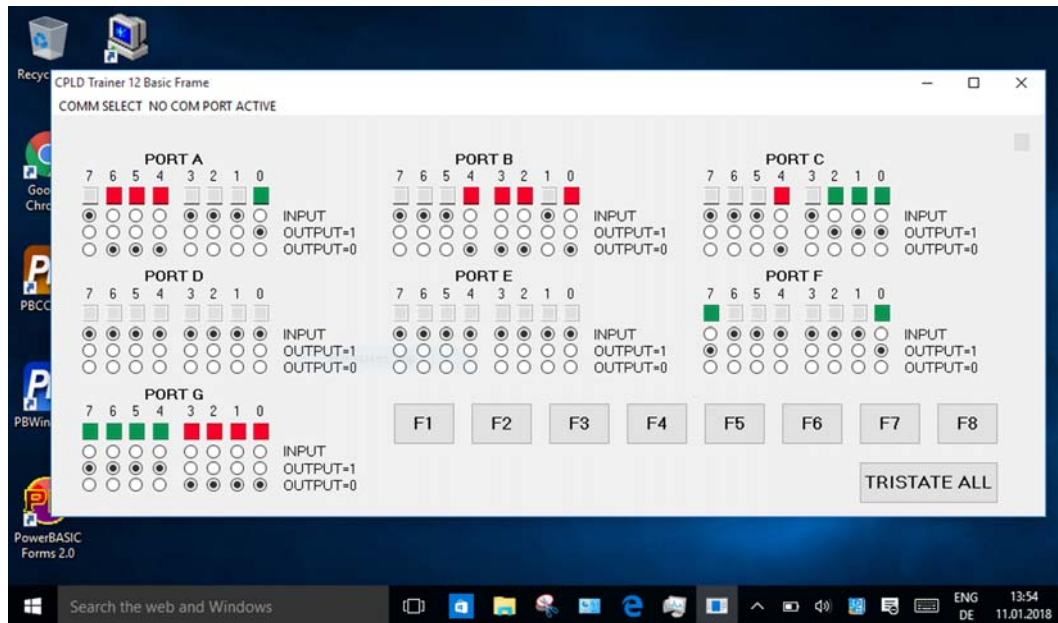


Abb. 1.30 Eine weiteres Bedienprogramm, dargestellt auf einem 7"-Tablet. Alle 7 Ports = 56 Bitpositionen des CPLD-Lehrgerätes 12 können angezeigt und – als Ausgänge – auf den jeweils gewünschten Signalpegel geschaltet werden. Die Anzeige wird live aktualisiert. Eine Hardwarelösung müsste 56 Zweifarben-LEDs und 56 Kippschalter mit drei Stellungen haben.

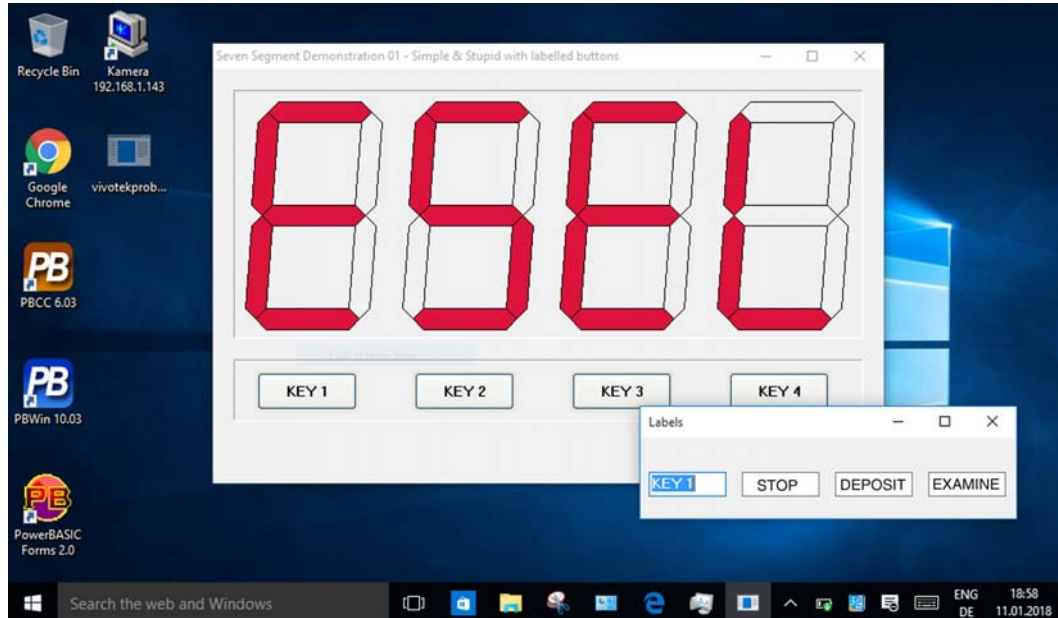


Abb. 1.31 Der PC emuliert die Peripherie. Hier eine vierstellige Siebensegmentanzeige mit vier Tasten. Die Tasten können beschriftet werden. Ein Mikrocontroller, der als Peripherieadapter arbeitet, fragt die Anzeigesignale ab und erregt die Tastensignale. Die Anzeige wird live aktualisiert.

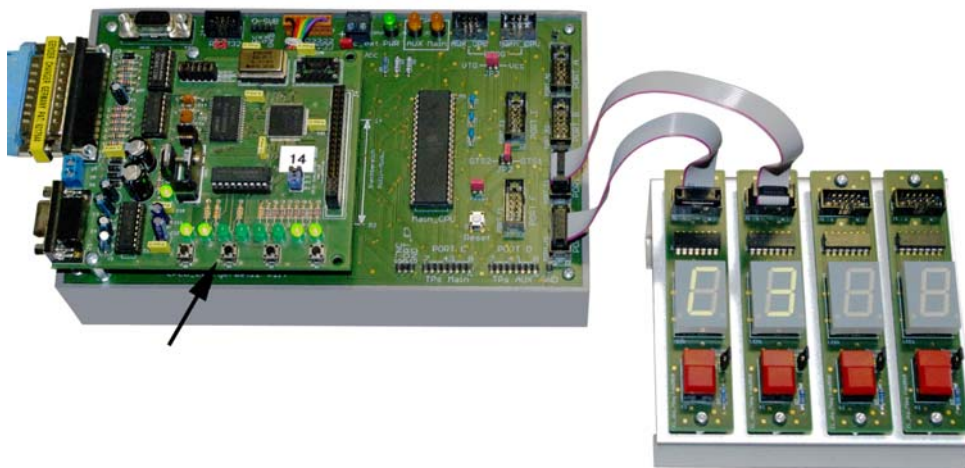


Abb. 1.32 Das ist die gleiche Peripherie als Hardware. Sie wird von der aufgesteckten CPLD-Übungsplatte (Pfeil) angesteuert. Das hier gezeigte CPLD-Lehrgerät 12 kann wahlweise als Peripherieadapter betrieben werden (und so die Emulation gemäß Abb. 1.31 unterstützen) oder die Signale der Übungsplatte 1:1 an die echte Peripherie weiterleiten.

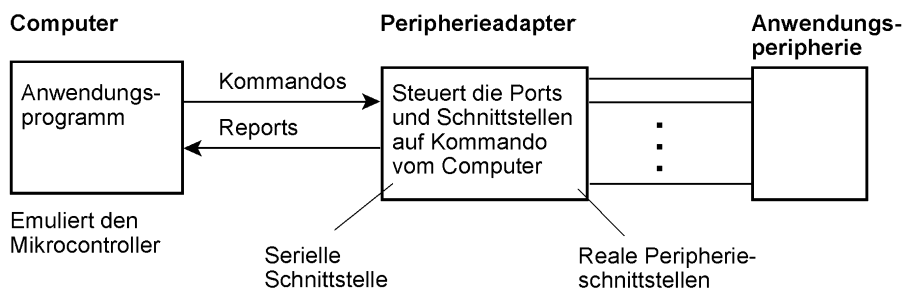


Abb. 1.33 Der Computer emuliert den Mikrocontroller. Die Peripherie wird angesteuert und abgefragt, als wäre der Mikrocontroller angeschlossen, nur langsamer.

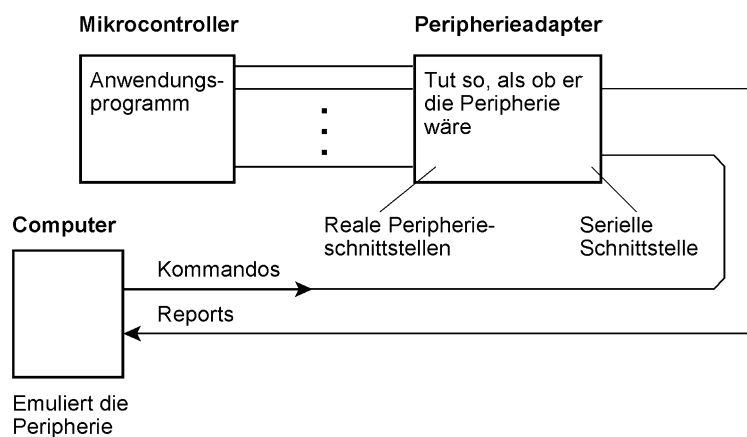


Abb. 1.34 Der Computer emuliert die Peripherie. Der Peripherieadapter verhält sich gegenüber dem Mikrocontroller wie die echte Peripherie, nur manchmal langsamer.

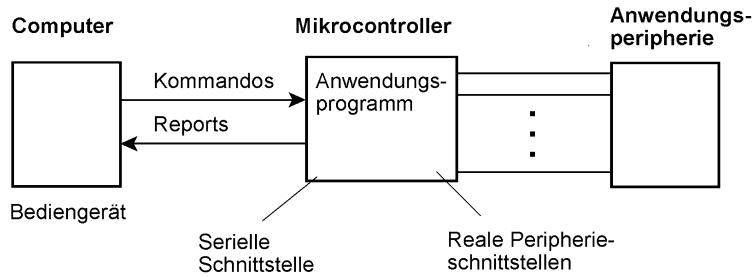


Abb. 1.35 Der Computer dient lediglich als Bedienkonsole (Plattformbedienung).

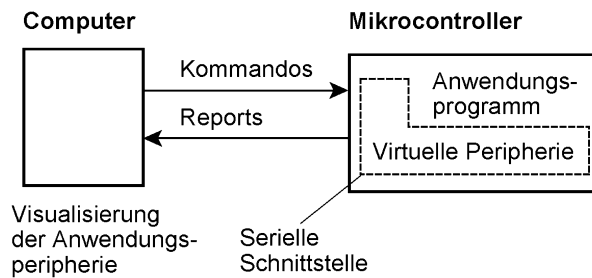
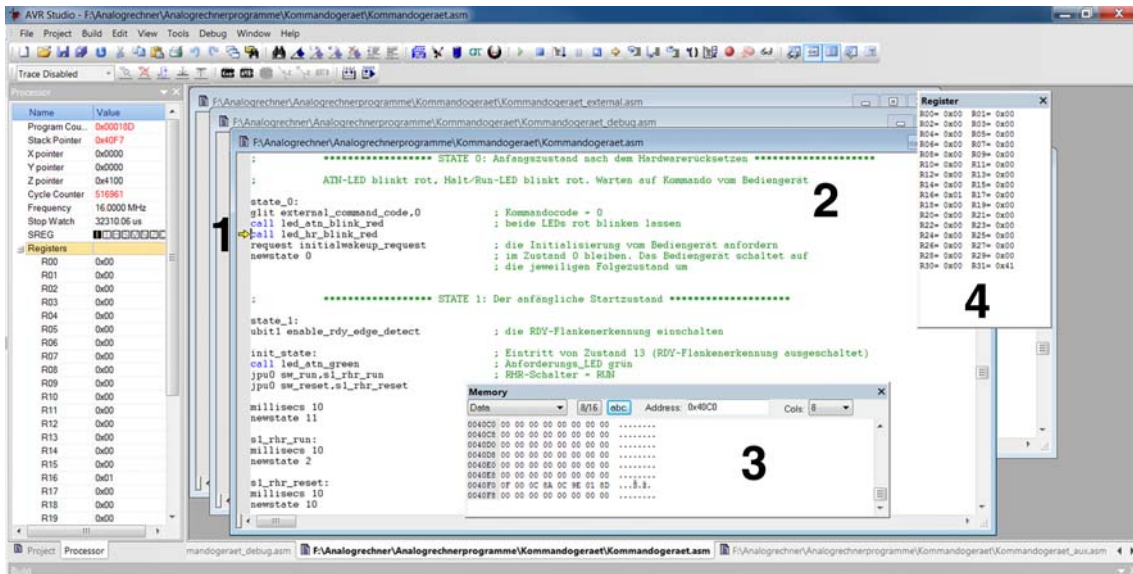


Abb. 1.36 Der Computer emuliert die Peripherie. Die E-A-Zugriffe werden im Mikrocontroller als virtuelle Zugriffe implementiert.



- 1 Der Pfeil zeigt auf den auszuführenden Befehl
- 2 Quelltext (Editorfenster)
- 3 Speicheranzeige
- 4 Registeranzeige

Abb. 1.37 Programmsimulation in einer Entwicklungsumgebung. Man kann alles bis aufs Bit verfolgen und beeinflussen. Besonders anschaulich ist es aber nicht...

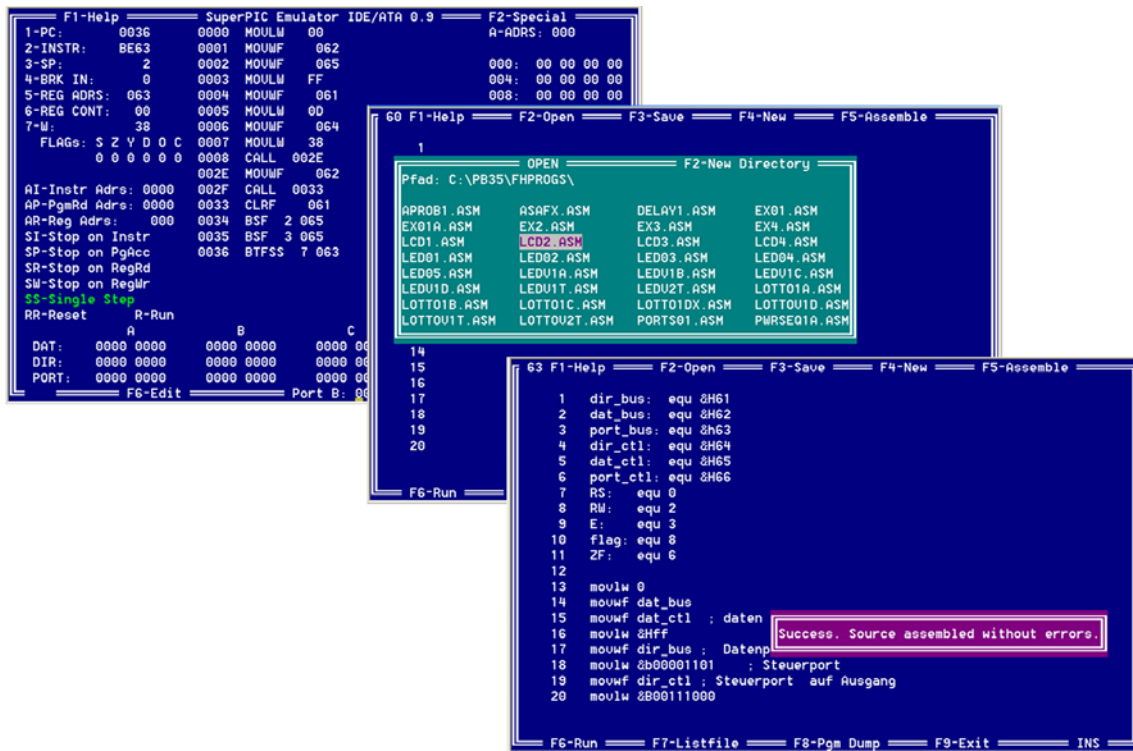


Abb. 1.39 Der PC emuliert den Mikrocontroller. Hier ist der PC klein und der Mikrocontroller einfach. Betriebssystem: DOS. Alles bleibt überschaubar. Das Anwendungsprogramm kann direkt mit der Hardware zusammenwirken. Deshalb ist der Emulator ebenso schnell wie ein echter Mikrocontroller dieser Leistungsklasse. Es geht auch ohne grafischen Bildschirm... Die Anzeigen von oben nach unten: Programmabau und Debugging – Dateiauswahl – Quellprogrammeingabe. Der Assembler ist eingebaut.



Abb. 1.40 Ein kleiner PC mit einer improvisierten Übungsplattform. Die Peripherie ist als Hardware vorhanden, der Mikrocontroller wird emuliert. Der Peripherieadapter ist in das PC-Gehäuse eingebaut.

1.4 Systemkonfigurationen

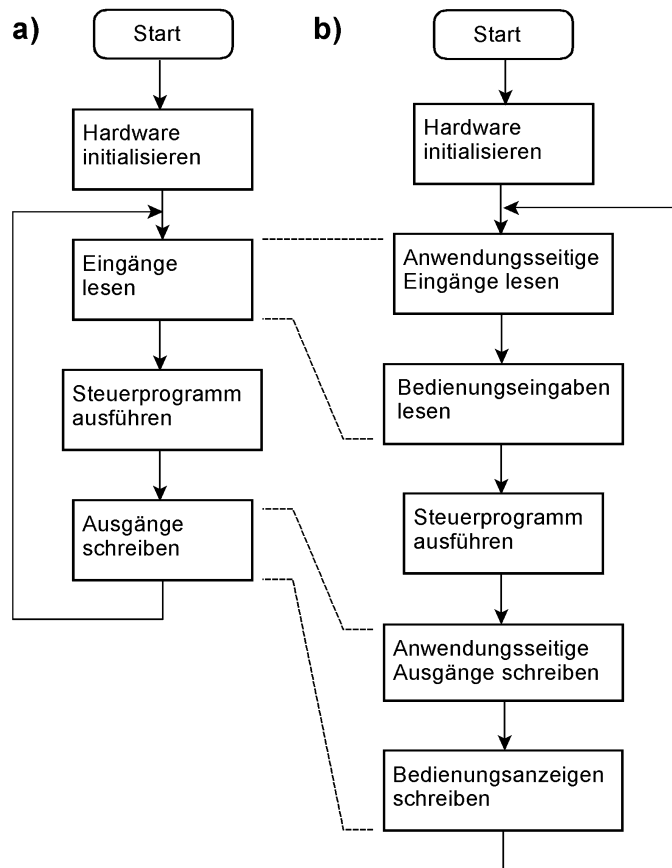


Abb. 1.41 Die Endlosschleife (Hauptsteuerschleife, Main Control Loop) ist die typische Grundstruktur der Programme in Embedded Systems.

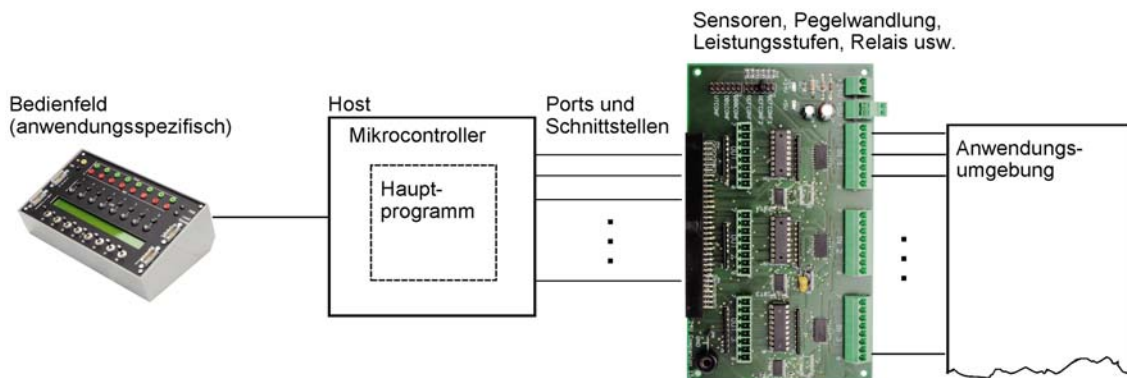


Abb. 1.42 Das Stand-Alone-System ist ein einfache Anwendungslösung auf Grundlage eines Mikrocontrollers. Er steuert alles. Die anderen Funktionseinheiten sind passive Einrichtungen (ohne jegliche Software) oder wirken als Slaves.

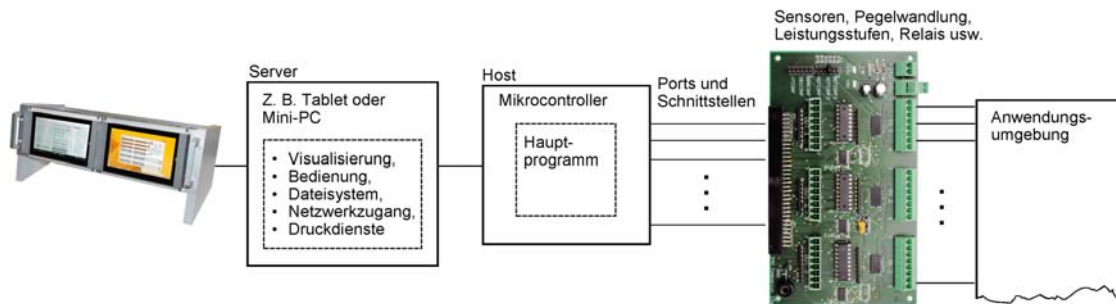


Abb. 1.43 Der Grundgedanke des Client-Server-System besteht darin, einen vollwertigen Computer an den Mikrocontroller anzuschließen. Dieser Computer kann die vielfältigsten Aufträge erledigen. Das kann sich bereits dann rechnen, wenn es nur um die Bedienung und Anzeige (Visualisierung) geht. Die eigentliche Anwendungssoftware läuft jedoch nach wie vor auf dem Mikrocontroller.

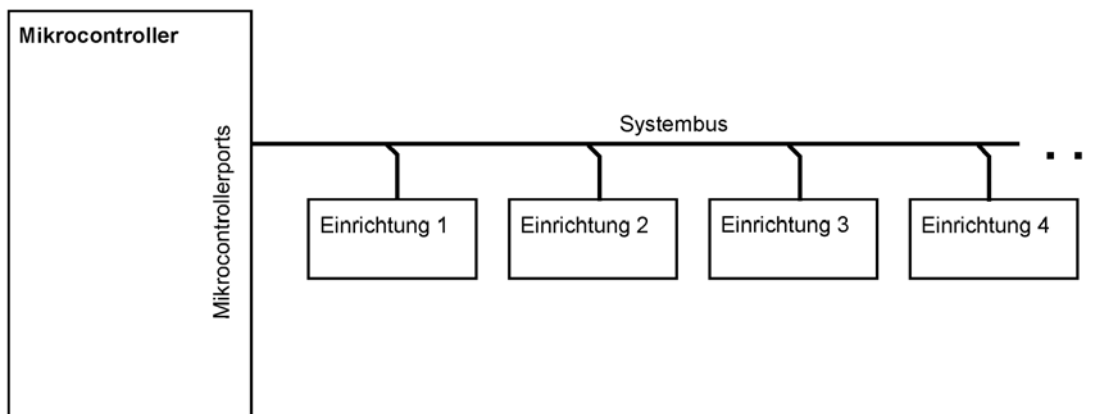


Abb. 1.44 Die Arbeit aufteilen. Hauptsteuerschleifen mit Auftragsvergabe. a) Client-Server, b) Master-Slave.

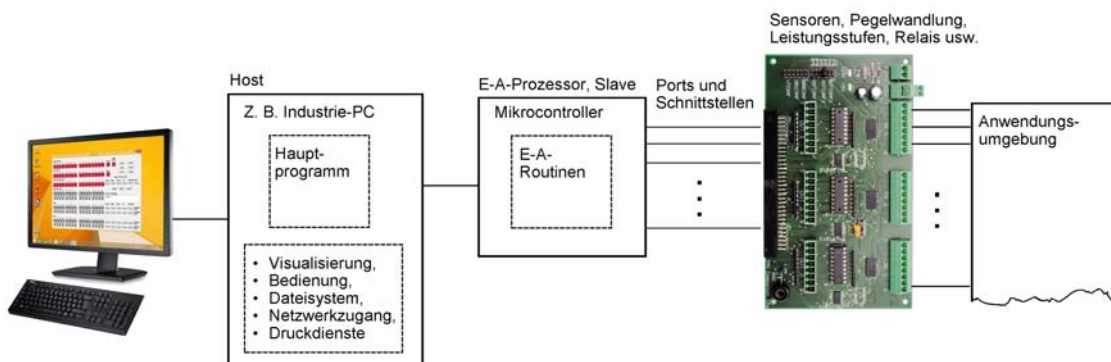


Abb. 1.45 Das Master-Slave-System entspricht der Konfiguration von Abb. 1.8. Der Mikrocontroller wirkt hierbei lediglich als programmierbare E-A-Einrichtung, die Kommandos oder heruntergeladene Programme ausführt.

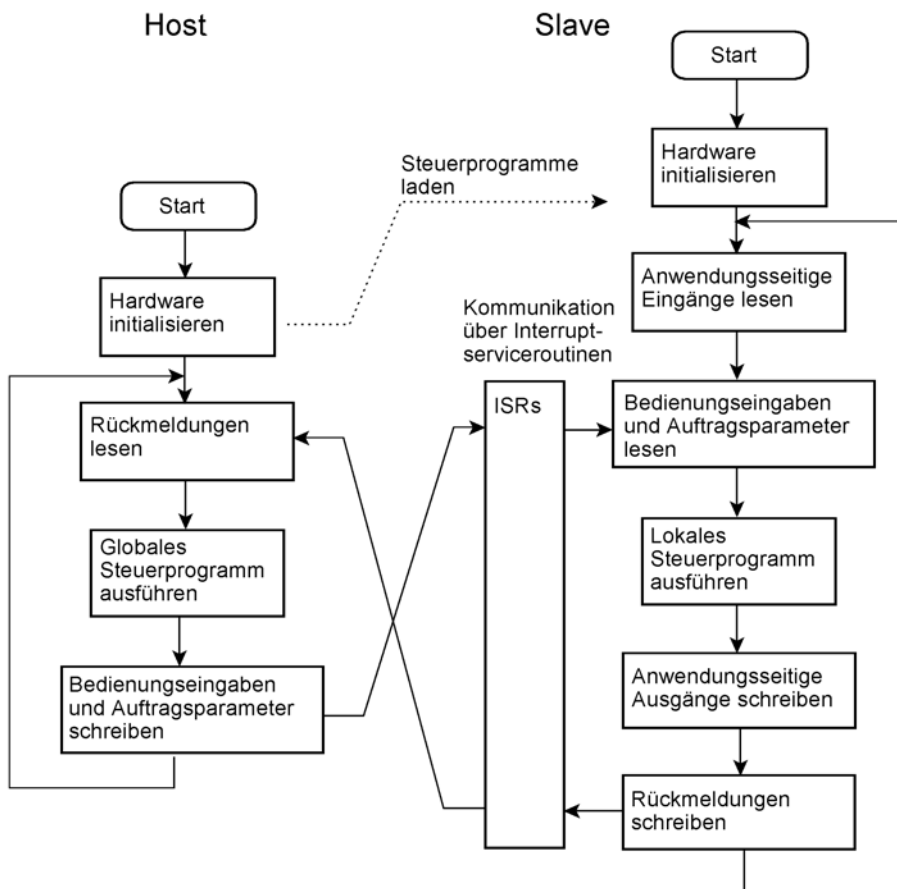


Abb. 1.46 Master-Slave-Organisation mit autonomer Programmausführung (Steuerschleife) im Mikrocontroller.

1.5 Mikrocontrollerschnittstellen ausnutzen

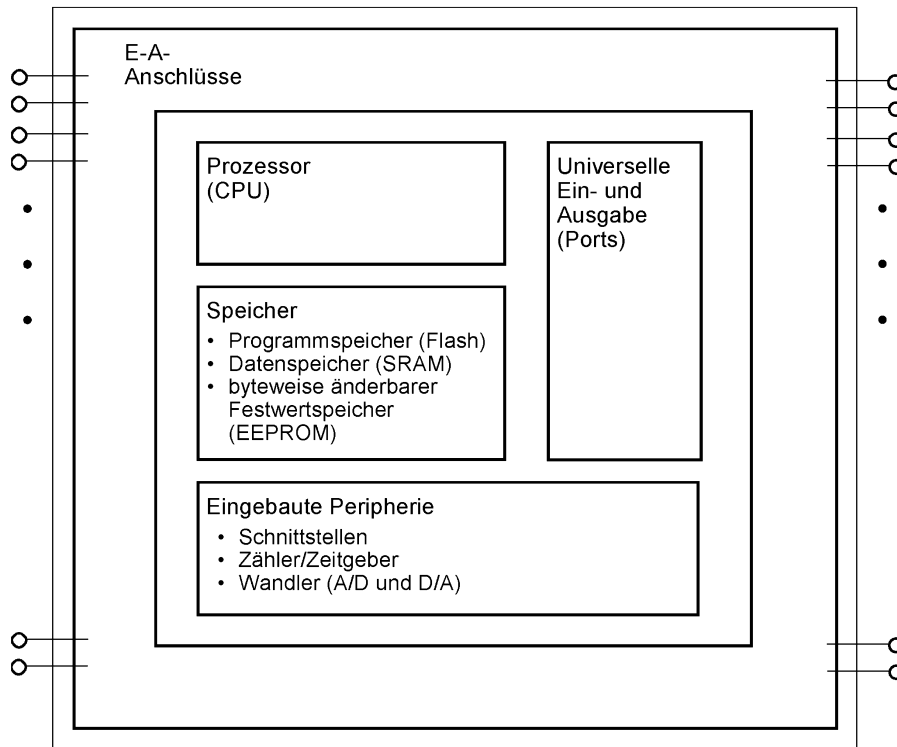


Abb. 1.47 Der Mikrocontroller. Die meisten Pins stehen zur Verfügung, um Signale der Anwendungsumgebung anzuschließen.

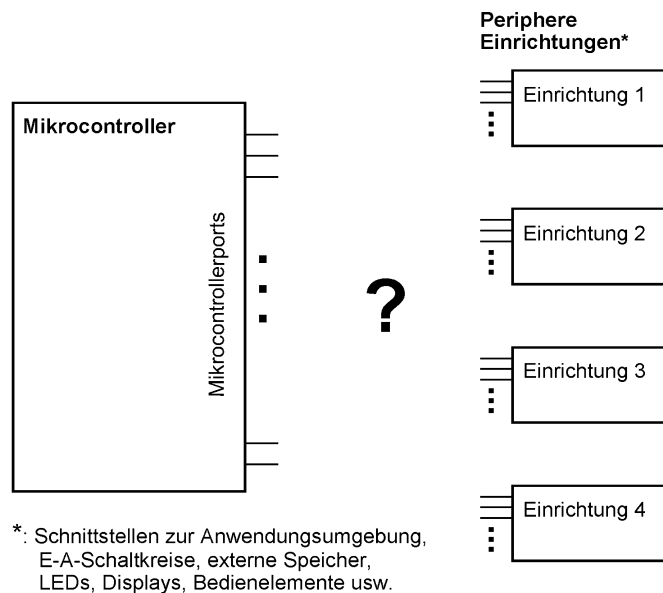


Abb. 1.48 Die grundsätzliche Entwurfsaufgabe.

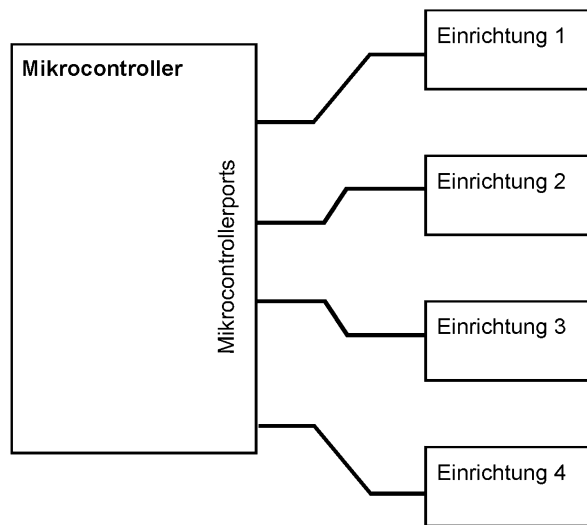


Abb. 1.49 Die Sterntopologie. Alle Einrichtungen werden einzeln angeschlossen.

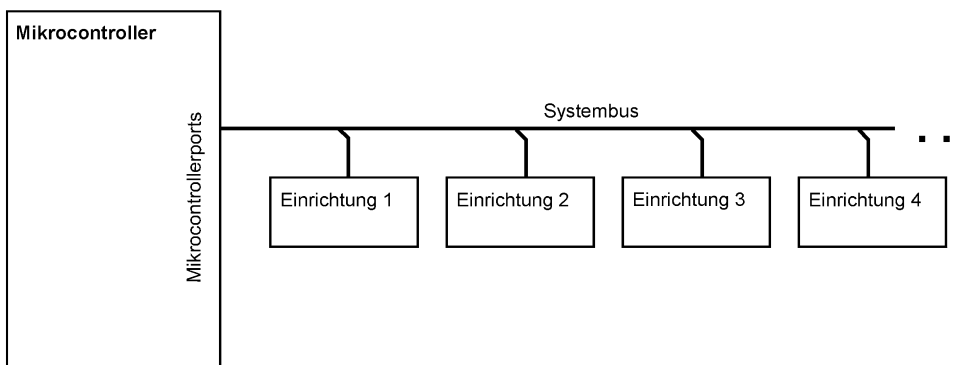


Abb. 1.50 Die Bustopologie. Alle Einrichtungen sind an gemeinsame Signalwege angeschlossen.

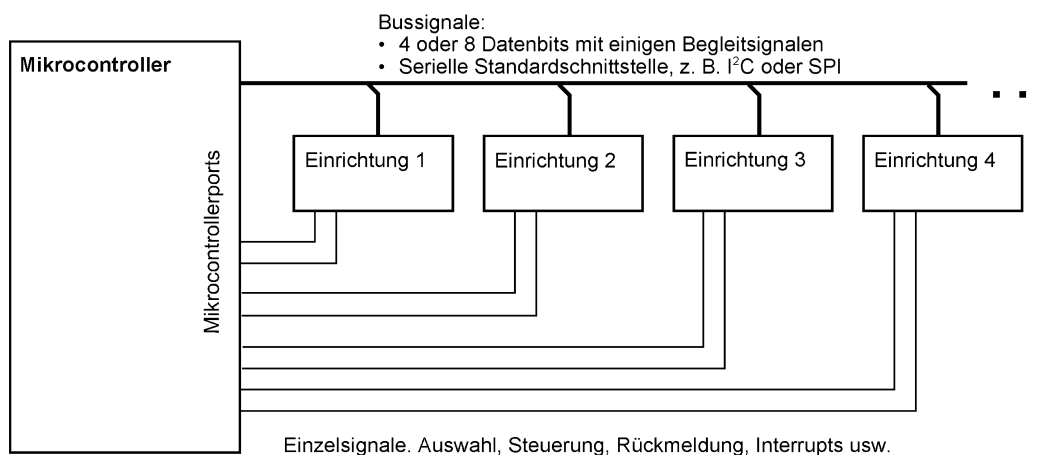


Abb. 1.51 Der typische Bus der Mikrocontrollerpraxis ist ein schmaler Signalweg, der bedarfsweise mit Einzelsignalen ergänzt wird.

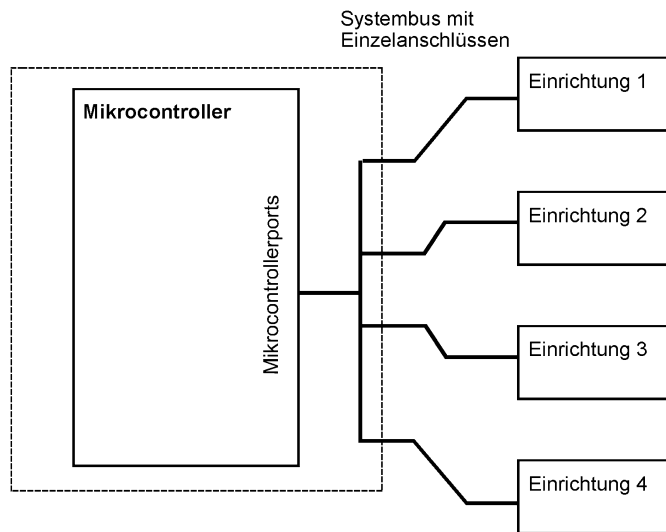


Abb. 1.52 Der Bus, der wie ein Stern aussieht. Was die Bustopologie kennzeichnet, sind die gemeinsamen Signalwege. Man muss sie nicht von Einrichtung zu Einrichtung weiterschleifen, sondern kann die Einrichtungen auch über Stichleitungen einzeln anschließen.

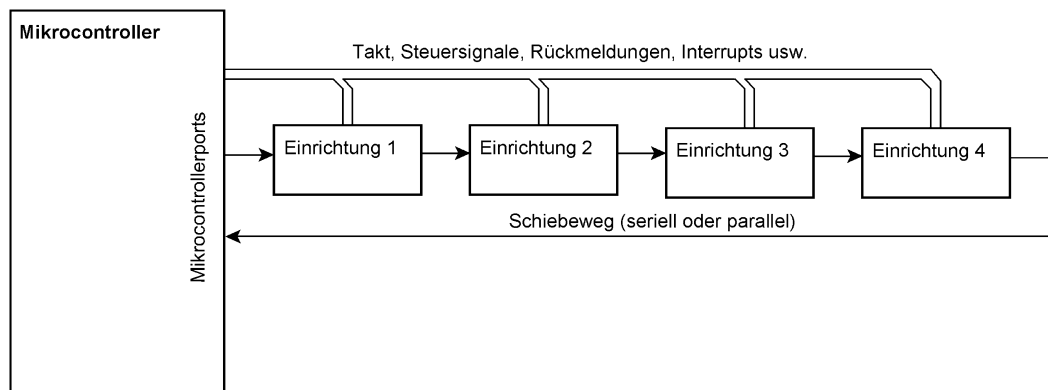


Abb. 1.53 Die Kettentopologie (Schieberegisterprinzip, Daisy Chain).

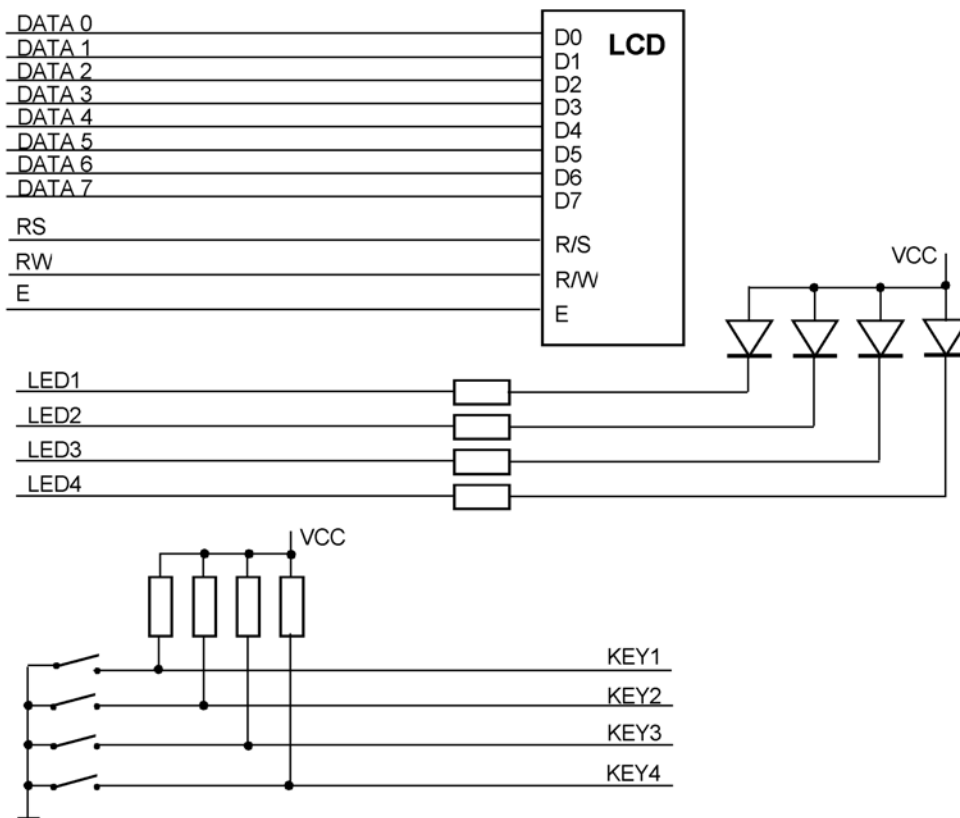


Abb. 1.54 Eine Bedientafel. Sie ist an einen Mikrocontroller anzuschließen.

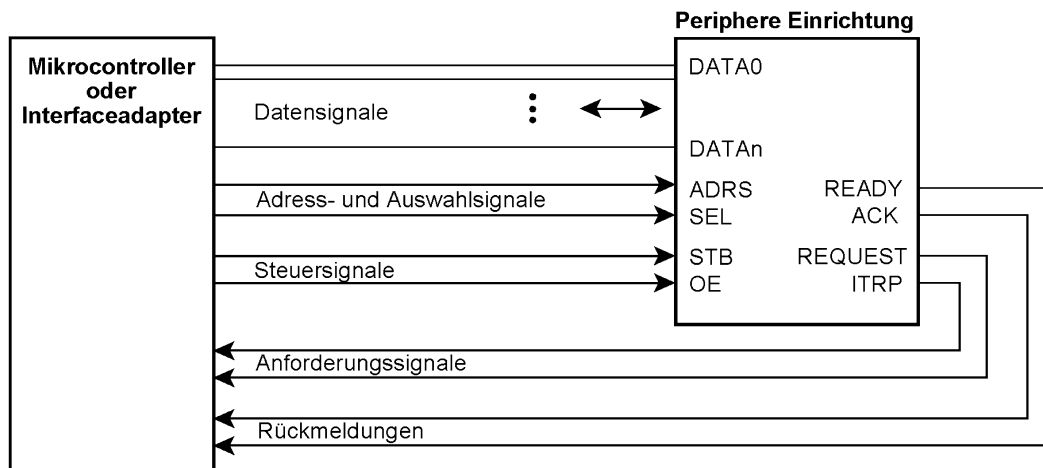


Abb. 1.55 Verschiedene Arten von Interfacesignalen.

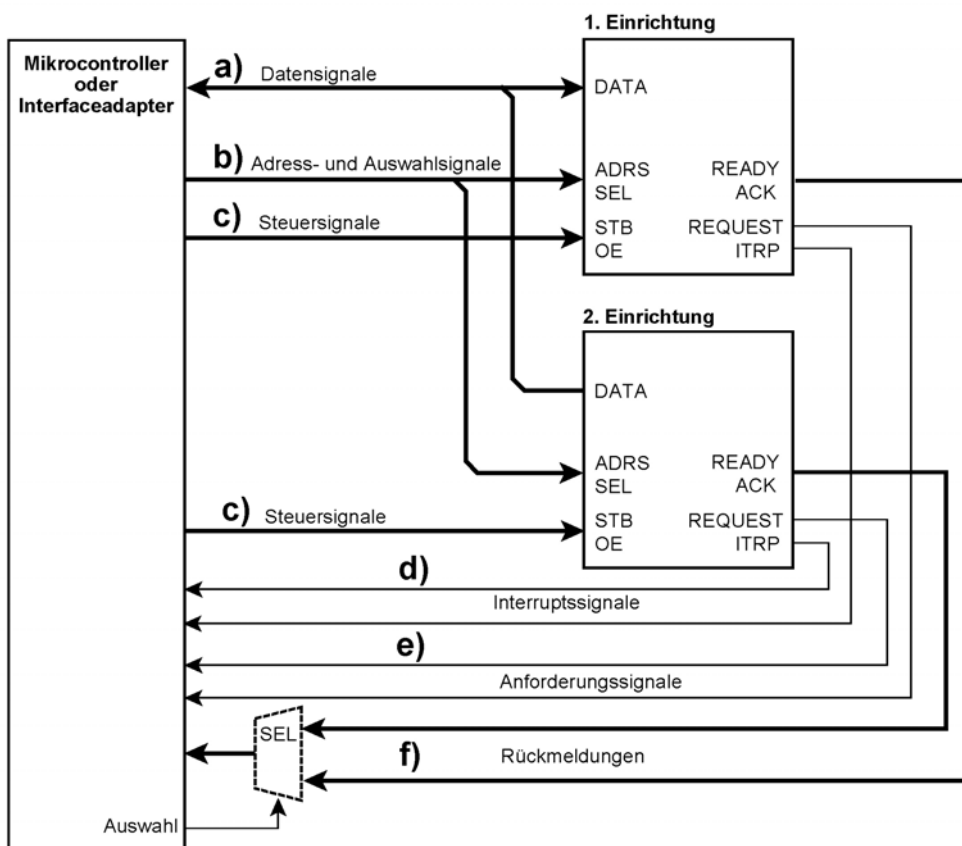


Abb. 1.56 Die Mehrfachnutzung von Interfacesignalen.

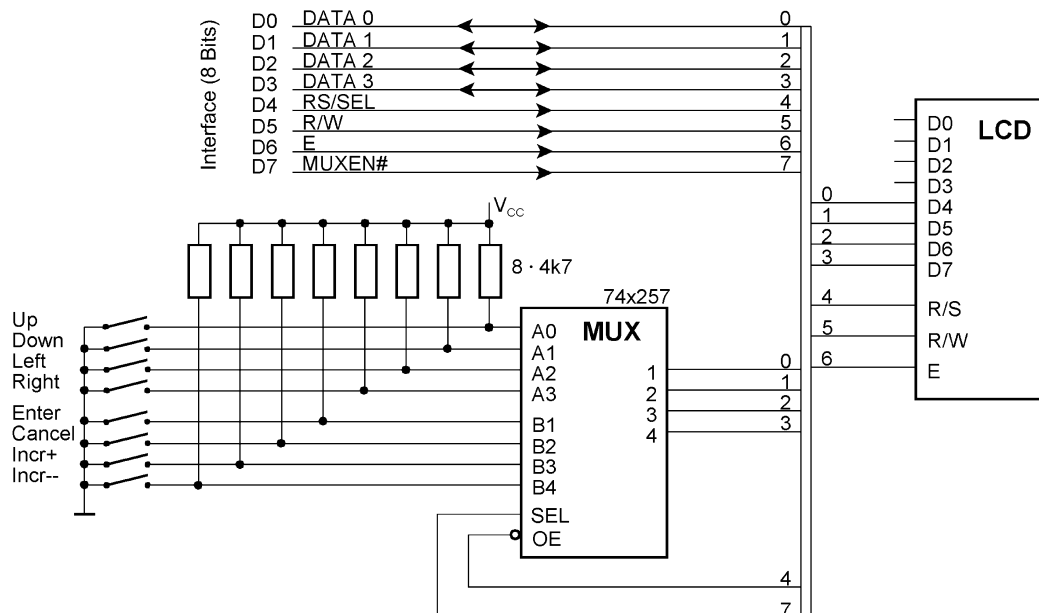


Abb. 1.57 Ein Praxisbeispiel: die LCD-Bedientafel Typ 1.

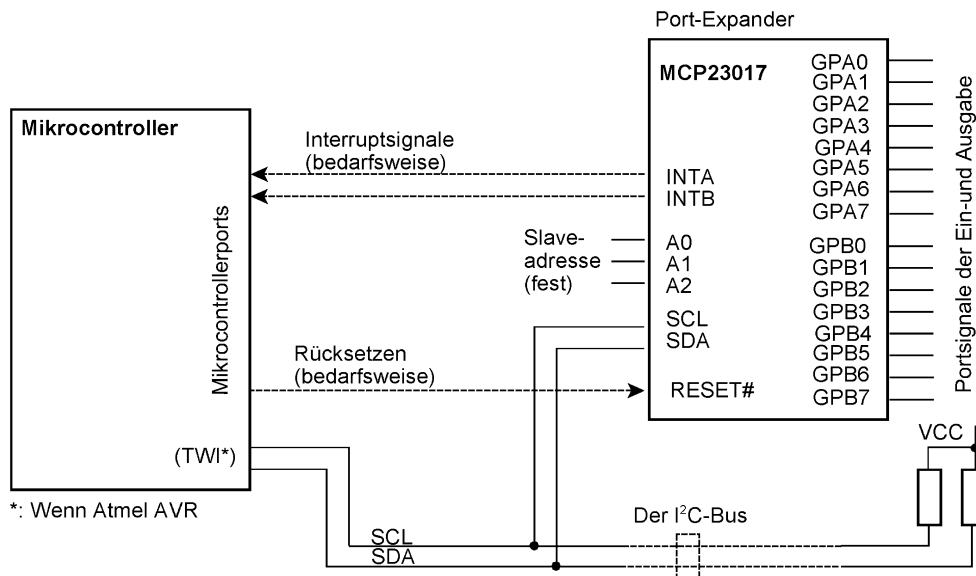


Abb. 1.58 Erweiterungsschaltkreis (Port Expander) mit I2C-Schnittstelle.

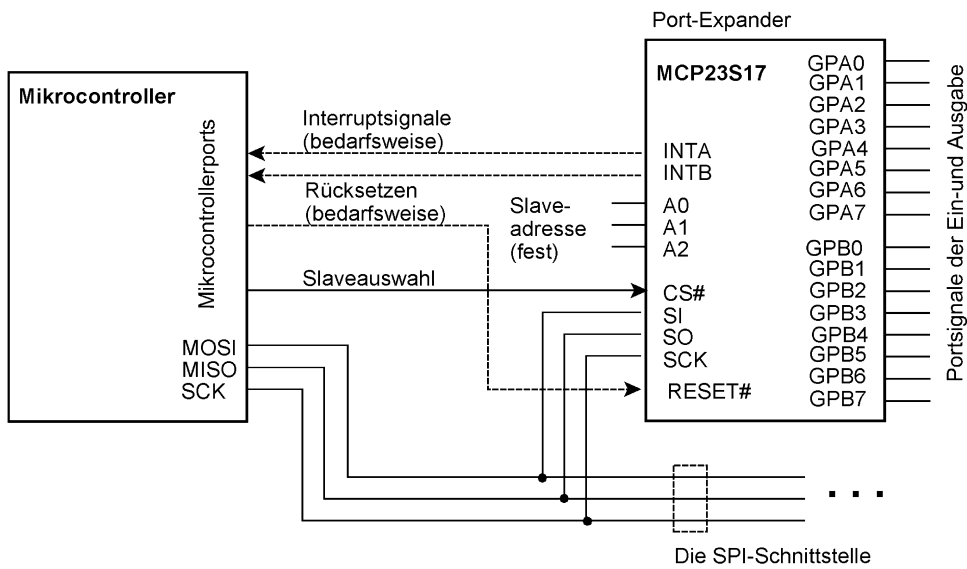


Abb. 1.59 Erweiterungsschaltkreis (Port Expander) mit SPI-Schnittstelle.

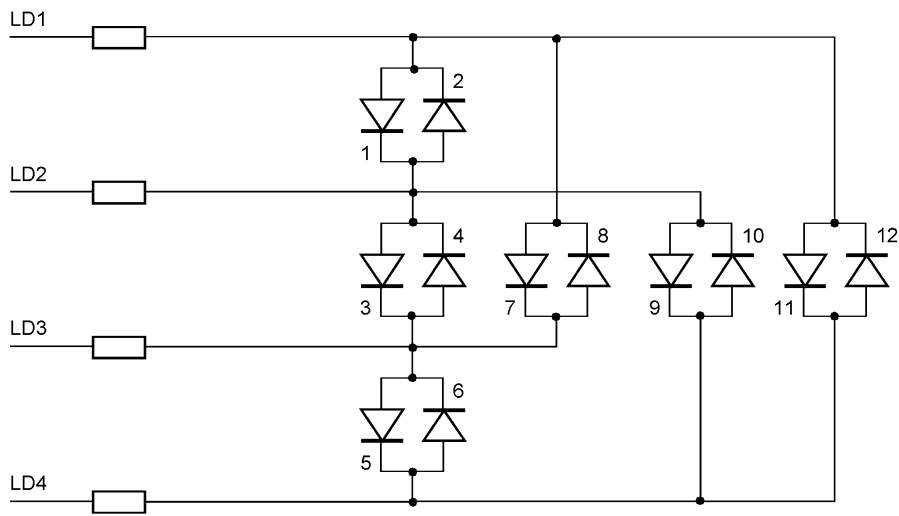


Abb. 1.60 Mit nur 4 Signalen 12 LEDs zeitmultiplex ansteuern.

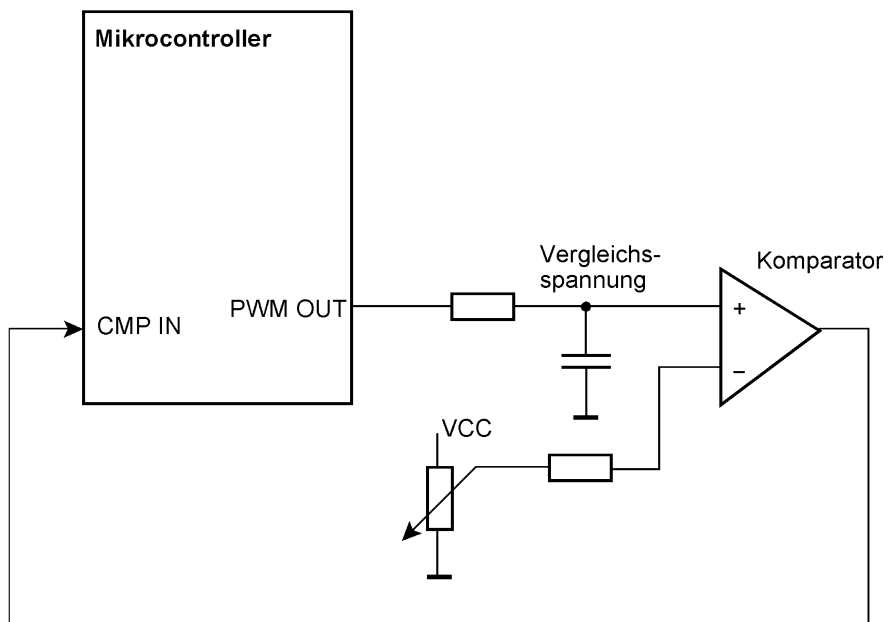


Abb. 1.61 Analog-Digital-Wandlung mit einfachen Mitteln.

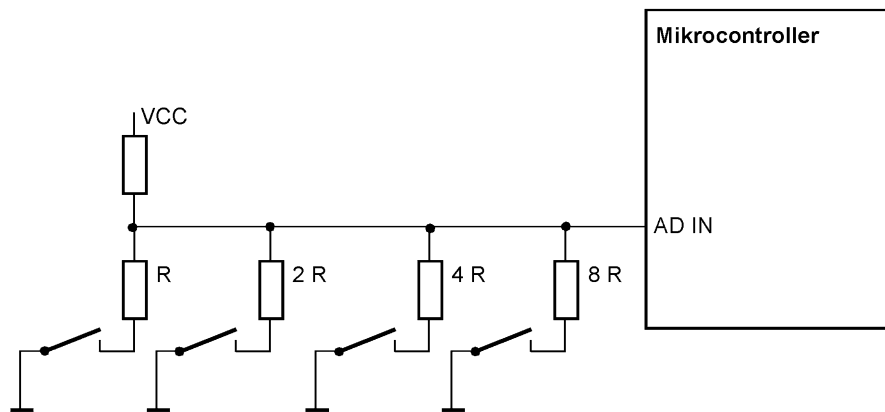
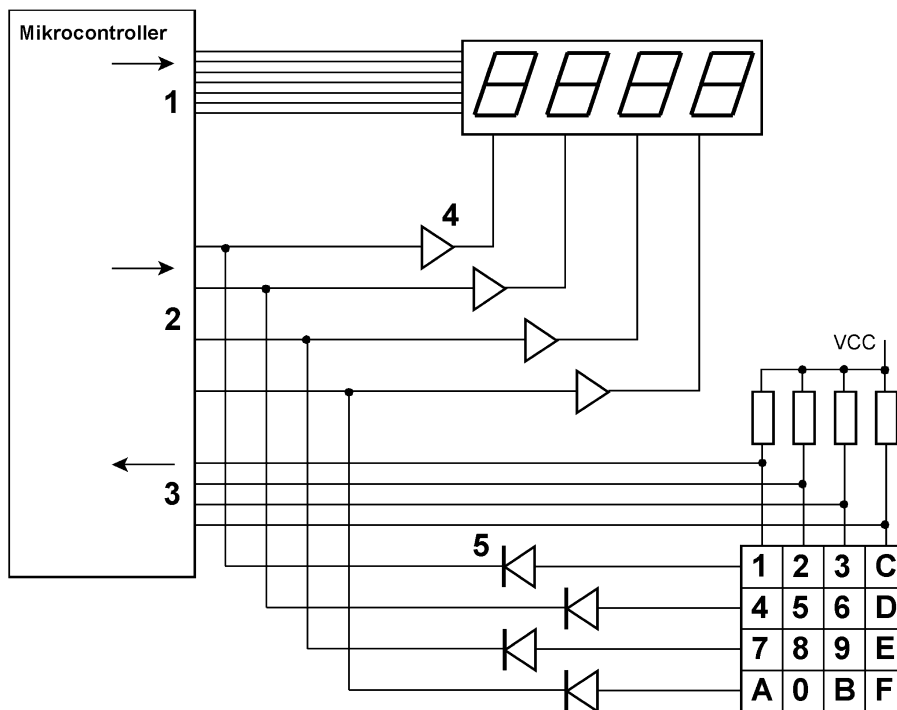


Abb. 1.62 Mehrere Tasten über einen einzigen Mikrocontrolleranschluß abfragen.



- | | | | |
|---|--|---|--|
| 1 | Segmentsignale | 4 | Digittreiber |
| 2 | Zifferstellen (Digit)- und Zeilensignale | 5 | Sperrdioden. Verhindern, dass bei Betätigung von zwei und mehr Tasten inaktive Zeilensignale auf Low gezogen werden. |
| 3 | Spaltensignale | | |

Abb. 1.63 Signale mehrfach nutzen. Hier werden eine Siebensegmentanzeige angesteuert und ein Tastenfeld abgefragt.

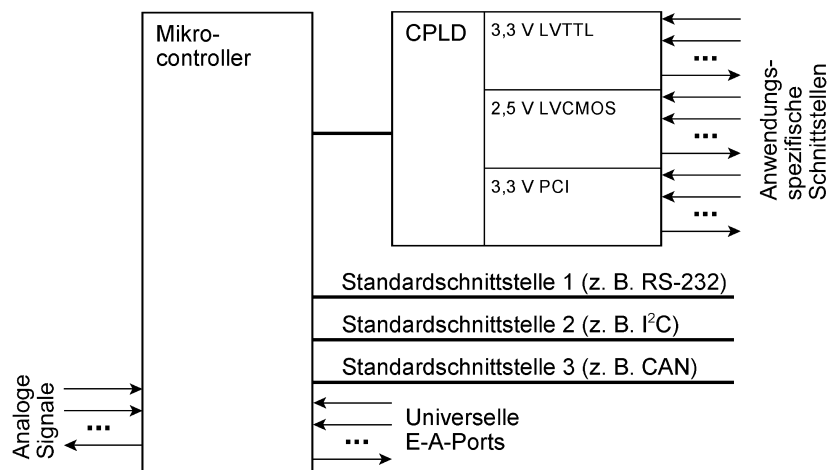


Abb. 1.64 Mikrocontroller und CPLD im Verbund. Höher entwickelte CPLDs unterstützen mehrere Pegelspezifikationen.



Abb. 1.65 Die Realzeitplattform 13. Links die Mikrocontrollerplatine, rechts die CPLD-Platine. Beide können auch unabhängig voneinander genutzt werden.

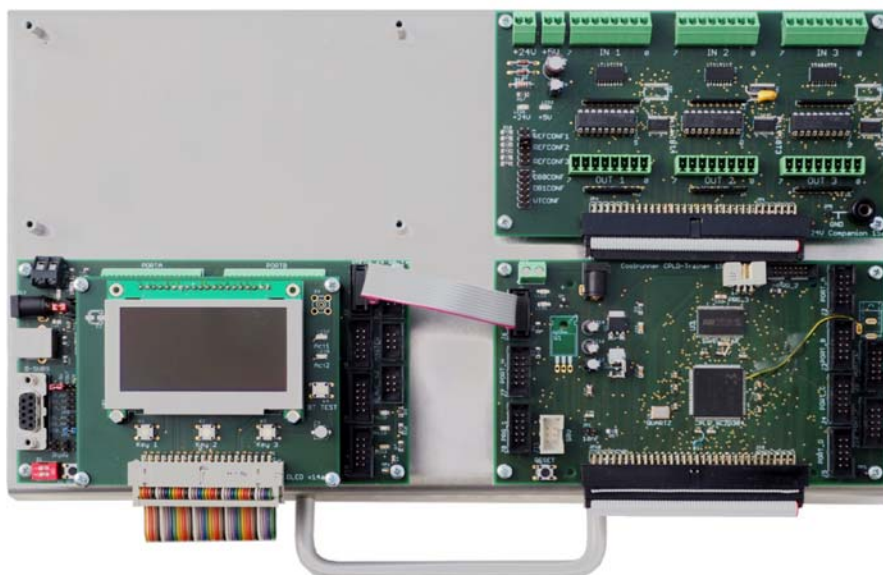


Abb. 1.66 Links ein Xmega-Trainer, rechts daneben der CoolRunner CPLD Trainer 15, darüber der 24-V Companion 15a.



Abb. 1.67 Das CPLD-Lehrgerät 12. Die Plattform (untere Platine) enthält zwei Mikrocontroller, das CPLD-Modul wird aufgesteckt.



Abb. 1.68 Kleine steckbare Module erlauben es, beliebige Verbundlösungen mit CPLDs und Mikrocontrollern zu verwirklichen.

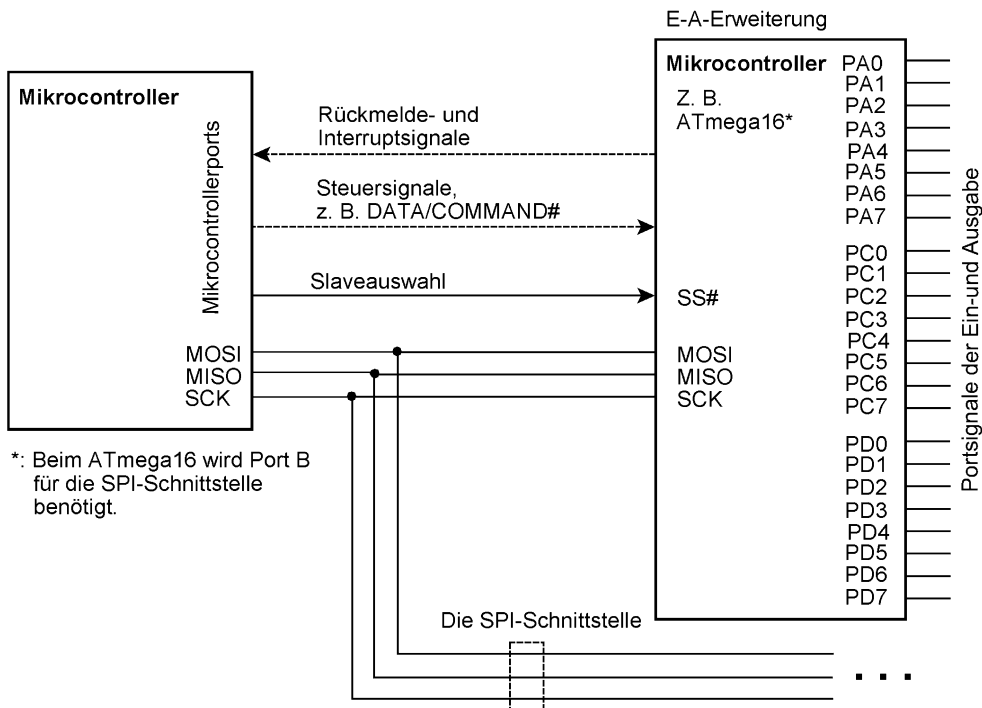


Abb. 1.69 Der Mikrocontroller als Port-Erweiterungsschaltkreis.

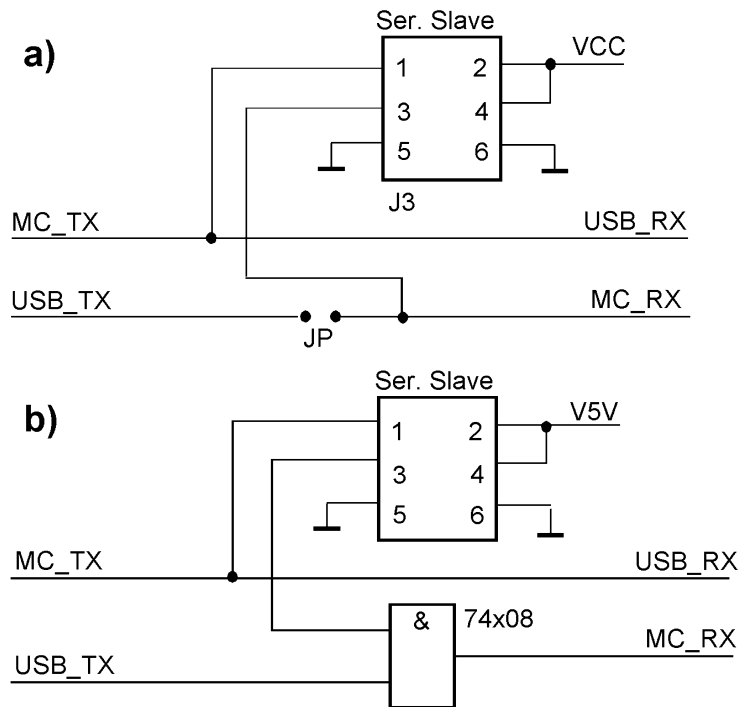


Abb. 1.70 Hier kann eine serielle Schnittstelle als 5-V-Slave oder für die USB-Signalisierung konfiguriert werden. Einzelheiten (Pegelwandlung, Pull-up-Widerstände usw.) sind nicht dargestellt. MC_TX = Sendesignal vom Mikrocontroller, MC_RX = Empfangssignal zum Mikrocontroller.

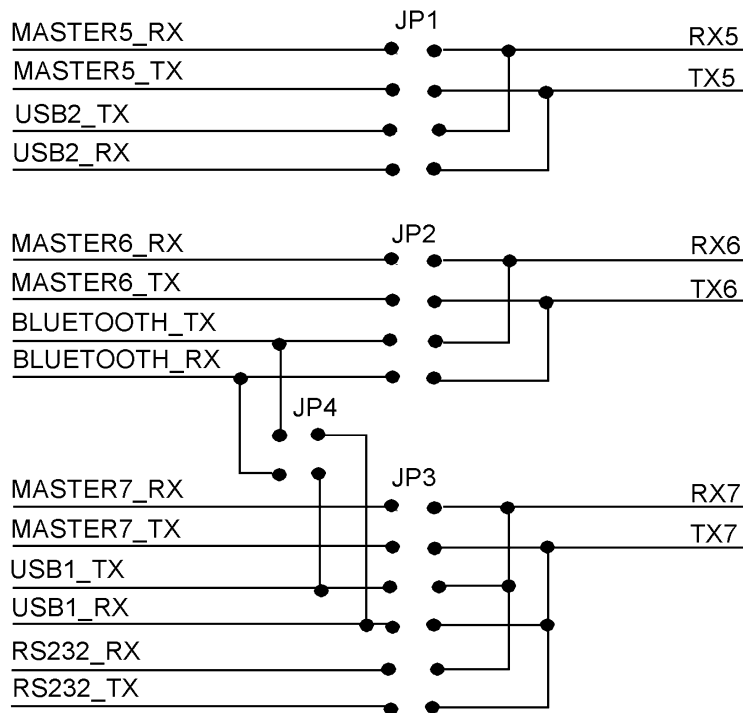


Abb. 1.71 Herkömmliche Konfigurationsvorkehrungen für drei serielle Schnittstellen und insgesamt sieben Betriebsarten.



Abb. 1.72 Konfigurationsmenüs, angezeigt auf dem OLED-Display einer Aufsteckplatine. S4 bis S7 sind serielle Schnittstellen, die so konfiguriert werden können, wie es angezeigt ist. Die aktuelle Konfiguration kann im EEPROM gespeichert werden.

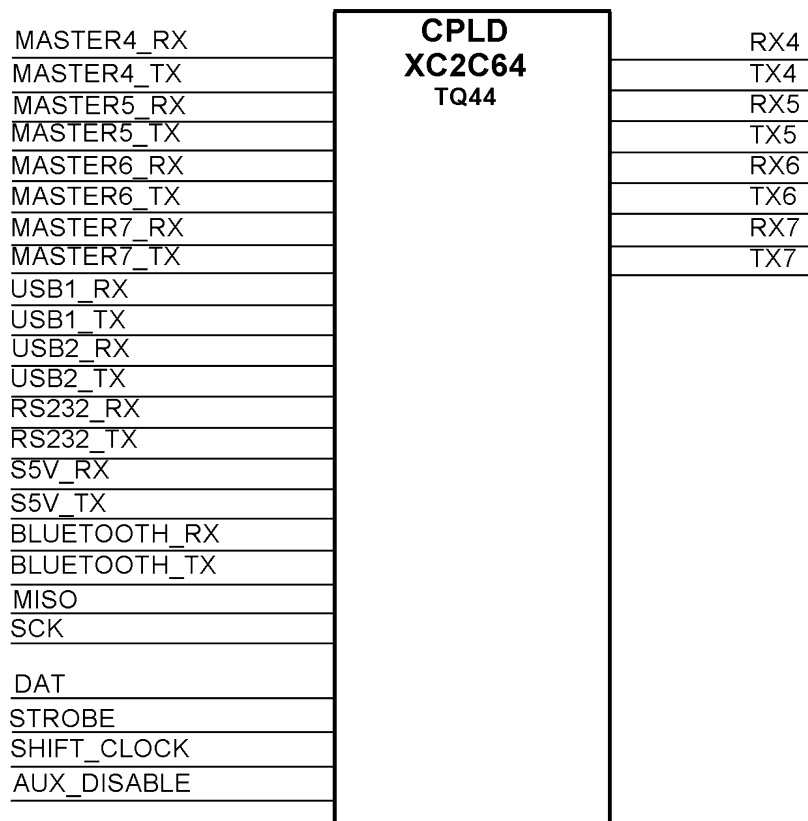


Abb. 1.73 Die Signalwegumschaltung ist in einem CPLD untergebracht. Hier geht es um 4 Schnittstellen und insgesamt 10 Betriebsarten. Links oben die Signale der Betriebsarten, rechts die Schnittstellensignale des Mikrocontrollers. Programmseitige Konfigurationseinstellung über ein einfaches Schieberegisterinterface (die vier Signale links unten). Es sind 11 Konfigurationsbits einzuschieben.