

6. Bedienung und Anzeige

6.1 Inbetriebnahme und Fehlersuche



Abb. 6.1 Bedienfelder historischer Computer. Eine kleine Auswahl (IBM 701, DEC PDP8/e, CDC 160, SDS Sigma 9).

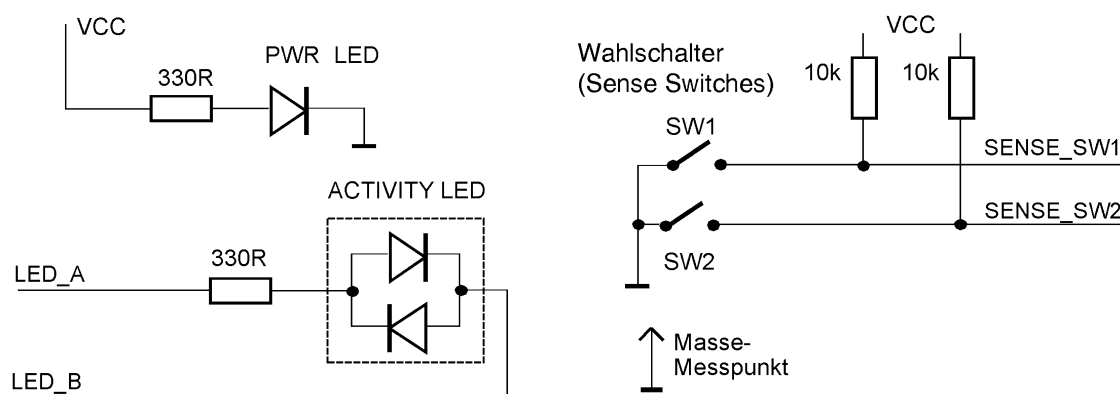


Abb. 6.2 Die typische Grundausstattung eines Mikrocontrollermoduls.

| Hersteller | Akustische Signalisierung |
|----------------|--|
| AMI | Kurze und lange Piepstöne. Maximal 9 Töne hintereinander. |
| IBM | Kurze und lange Piepstöne. Wenn o.k., dann eine einziger kurzer Ton. Dauerton bei einigen schweren Fehlern. Sonst maximal 4 Töne hintereinander. |
| Phoenix, Intel | Zumeist Folgen von Piepstönen. Beispiel: 1 Piepston – Pause – 3 Piepstöne – Pause – 3 Piepstöne – Pause – 1 Piepston. In der Dokumentation wird das beschrieben als "1-3-3-1". |

Tabelle 6.1 Akustische Fehlersignalisierung der Personalcomputer (historische Beispiele).

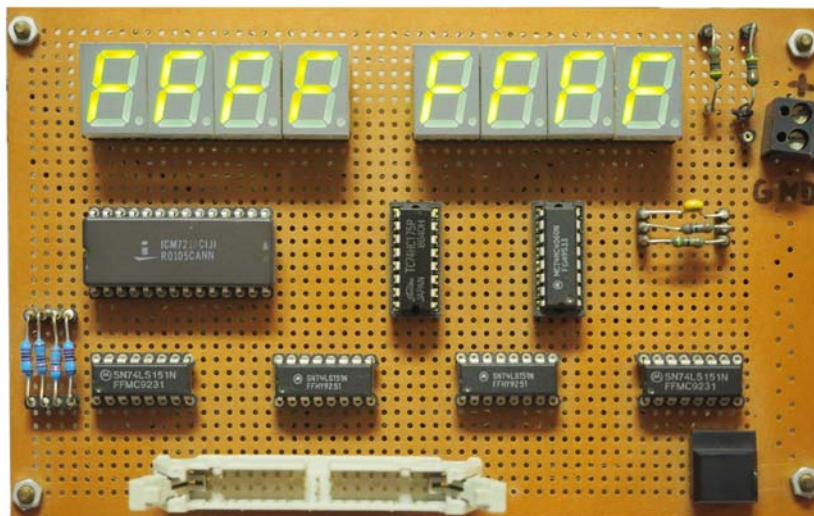


Abb. 6.3 Dieser sog. Logikmonitor ist ein Anzeigemodul in herkömmlicher Bauweise – ganz ohne Mikrocontroller. Er kann 32 Digitalsignale über Multiplexer abfragen und hexadezimal anzeigen. Die jeweils zu prüfende Platine muss mit einem entsprechenden Steckverbinder bestückt werden, oder die Signale werden wie beim Logikanalysator einzeln angeschlossen.

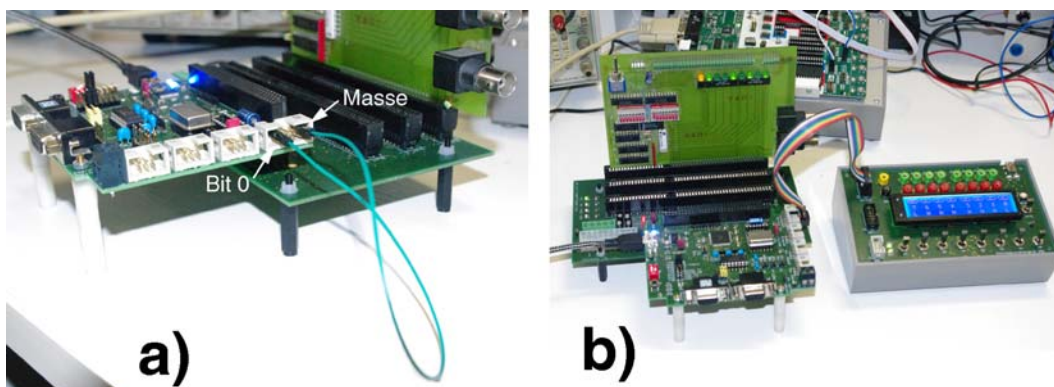


Abb. 6.4 Testauswahl mittels Portabfrage. a) Ein Portsignal wird auf Masse gezogen. b) Angeschlossene Kippschalter. Hier das Kippschalterkleingerät 15.

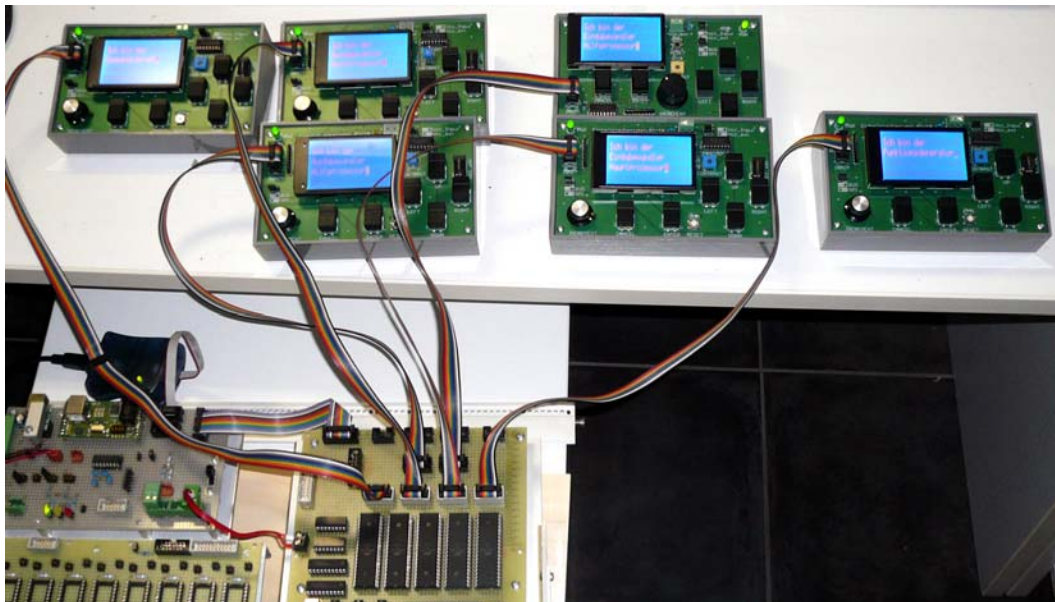


Abb. 6.5 Hier wird ein Mehrprozessorsystem mit insgesamt sechs Mikrocontrollern in Betrieb genommen. Am Anfang bekommt jeder eine eigene Bedienkonsole, so dass man auf einen Blick sehen kann, welcher arbeitet und welcher nicht.



Abb. 6.6 Sechs Bedienkonsolen sind etwas unübersichtlich. Nachdem die größten Fehler beseitigt sind, sollten zwei ausreichen. Sie wurden hier auf eine Frontplatte montiert, die zum Inbetriebnehmen und Fehlersuchen an der Maschine befestigt wird (Inbetriebnahmehilfsgerät).



Abb. 6.7 Zwei Anzeigebeispiele. Links die letzten vier Maschinenzustände. Ausgabe nach Art einer Protokollschreibmaschine (Eintragen in die unterste Zeile, nachdem der Text hochgerollt wurde). Rechts werden die Werte von acht Variablen hexadezimal angezeigt.

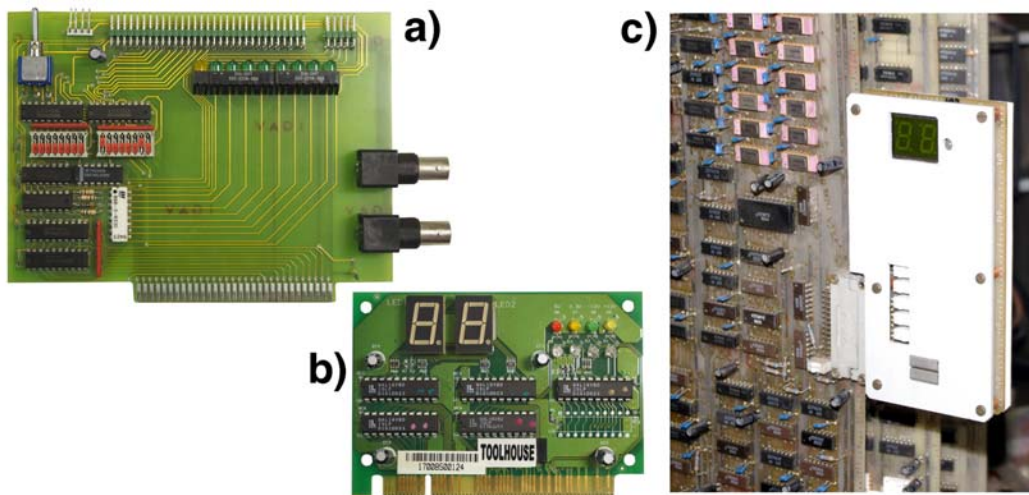


Abb. 6.8 Diagnoseadapter. a) Für den ISA-Bus. Adresse einstellbar, Ausgabe rücklesbar, Messpunkte für die Bussignale, Oszilloskopanschluss. b) Eine POST-Karte für den PCI-Bus. c) Ein gerätespezifischer Diagnoseadapter mit acht DIL-Schaltern zur Dateneingabe. Sie dienen zur Testauswahl und Ablaufsteuerung.

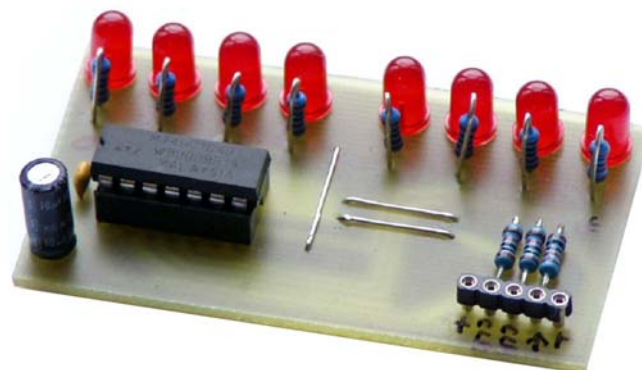


Abb. 6.9 Dieser einfache Diagnoseadapter braucht nur zwei Signale.

6.2 Anwendungslösungen



Abb. 6.10 Bedien- und Anzeigeelemente beschriften. Links Kippschaltergerät mit LCD-Anzeige. Beschriftungseingabe über Bedienterminal. Rechts ein Bedien- und Beschriftungsprogramm.

6.3 Lernen und Experimentieren

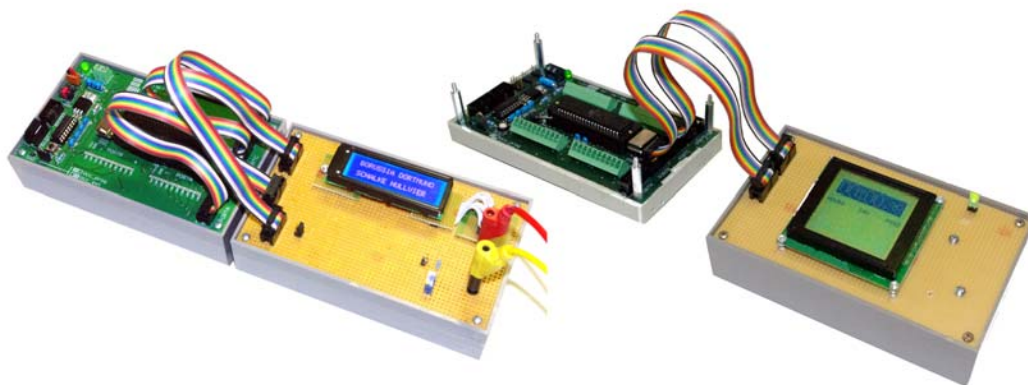


Abb. 6.11 Selbst aufgebaute Übungsstücke. Es muss nicht schön aussehen, aber halten

6.4 Computer als Bediengeräte

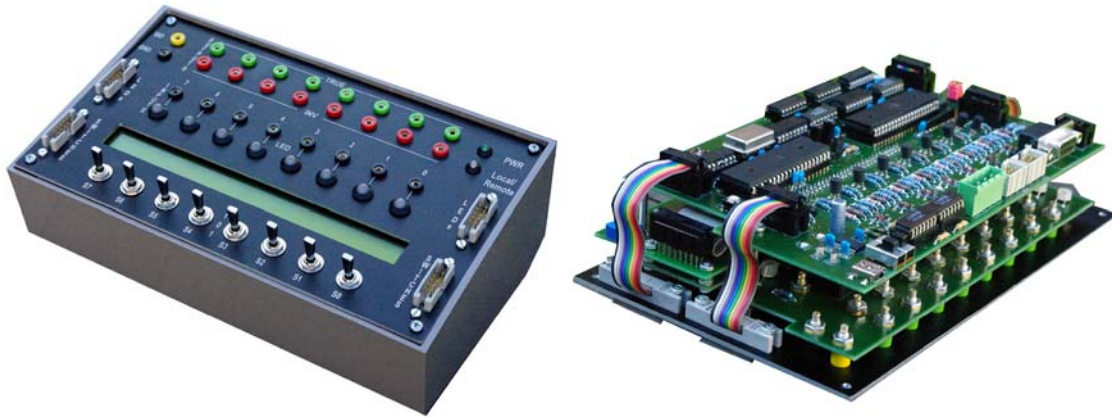


Abb. 6.12 Ein Kippschaltergerät von außen und innen. Der Aufwand ist offensichtlich.



Abb. 6.13 Kleine Computer kosten weniger als allein das Material einer anspruchsvollen Selbstbaulösung. Man kann sich sogar mehr als nur einen leisten...

| Bediencomputer | Bedien- und Anzeigeeinrichtungen |
|--|--|
| Mikrocontrollermodule | Beschränkt. Kleine Display-Module, wenige Tasten usw. Können aber anwendungsspezifisch ausgestaltet und erweitert werden, z. B. mit mehreren Displays. |
| Computermodule | Beschränkt. Displays mit höherer Auflösung und Touchscreen verfügbar. Anwendungsspezifische Ausgestaltung möglich, aber mit Arbeit verbunden. |
| Besonders kleine PCs (Netbooks, Mini-PCs, PC-Sticks, TV-Boxes, Tablets usw.) | Beschränkungen in Hinsicht auf Bildschirmgröße und Auflösung sowie auf anschließbare Bedienmittel (USB, Bluetooth). Beim Gestalten der Bedienoberflächen müssen die kleinen Bildschirme und die Besonderheiten der Bedienung (Touchscreen, keine Maus, Bildschirmtastatur usw.) berücksichtigt werden. |
| Personalcomputer | Keine Beschränkung |

Tabelle 6.2 Bediencomputer (1). Bedien- und Anzeigeeinrichtungen.

| Bediencomputer | Nutzung von Kommunikationsschnittstellen, Dateisystemen usw. |
|--|---|
| Mikrocontrollermodule | Möglich, aber arbeitsaufwendig. Open-Source-Software für Protokollstacks und Dateisystemunterstützung nutzen oder Einfachlösungen selbst entwickeln*. |
| Computermodule | An sich vorgesehen, Nutzung aber arbeitsaufwendig. Einarbeitung in komplexe Open-Source-Software (Linux, Skriptsprachen usw.), und das nur, um einfache Bedienfunktionen zu implementieren. |
| Besonders kleine PCs (Netbooks, Mini-PCs, PC-Sticks, TV-Boxes, Tablets usw.) | Keine Beschränkung, keine Mehrarbeit. Alle Standardlösungen nutzbar. |
| Personalcomputer | Keine Beschränkung, keine Mehrarbeit. Alle Standardlösungen nutzbar. |

*: Wobei sich aber das Problem der Kompatibilität und Portabilität ergibt. Ein einfaches Dateisystem liegt durchaus im Rahmen des Möglichen. Kann man aber solche Dateien auch woanders lesen oder erstellen¹?

Tabelle 6.3 Bediencomputer (2). Was ist zu tun, um Kommunikationsschnittstellen, Dateisysteme usw. zu nutzen?

1: Auch das lässt sich einrichten. Man muss nur wissen, wie...

| Bediencomputer | Arbeitsaufwand (Programmierung) |
|--|--|
| Mikrocontrollermodule | Maschinennahe Programmierung (Low-Level). Einfache, überschaubare Systemumgebung. Alles muss von Grund auf programmiert werden, auch die ganz elementaren Funktionen, wie die pixel- oder zeichenweise Bildarstellung, die Tastenabfrage usw. |
| Computermodule | Anwendungsorientierte Programmierung (High-Level). Komplexe Systemumgebung als Open Source. Hoher Einarbeitungsaufwand. Es kann aber vorkommen, dass auch elementare bzw. physische Funktionen selbst zu programmieren sind*. |
| Besonders kleine PCs (Netbooks, Mini-PCs, PC-Sticks, TV-Boxes, Tablets usw.) | Anwendungsorientierte Programmierung (High-Level). Komplexe Systemumgebung, aber komfortabel nutzbar. Alle Einzelheiten erledigt das System. Manche Entwicklungsumgebungen sind ausgesprochen teuer. Es ist für kleine Bildschirme und eingeschränkte Bedienmittel zu programmieren**. |
| Personalcomputer | Anwendungsorientierte Programmierung (High-Level). Komplexe Systemumgebung, aber komfortabel nutzbar. Alle Einzelheiten erledigt das System. Manche Entwicklungsumgebungen sind ausgesprochen teuer. |

*: Abhängig davon, was als Open Source bereitgestellt wird. Man muss es in der Menge der Software finden, und es muss praktisch auf Anhieb funktionieren. Sonst schreibt man es besser gleich selbst ...

** : Deswegen kann es mehr Zeit erfordern, die Bedienoberfläche zu konzipieren (gelingt nicht immer auf Anhieb). Beim Ausprogrammieren kann es sein, dass man sich mit Einzelheiten der Nachrichten (Messages) und Ereignisse befassen muss.

Tabelle 6.4 Bediencomputer (3). Der Arbeitsaufwand der Programmierung.



Abb. 6.14 Kleine Personalcomputer. Näheres in Tabelle 6.5.

| Abb. 6.14 | Typ | Abmessungen in mm | Bildschirmauflösung |
|-----------|--------------|---|-------------------------------|
| a) | Netbook | 250 • 180 (10,1 Zoll) bis 290 • 200 (11,6 Zoll)* | 1024 • 600 bis 1920 • 1280 |
| b) | Mini-PC | 120 • 120 • 24...50 | 1920 • 1080 |
| c) | PC-Stick | 100 • 40 • 12 | 1920 • 1080 |
| d) | TV Box o. ä. | 7 Zoll: 180 • 120 • 50 | 1280 • 800 |
| | | 8,9 Zoll: 220 • 150 • 60 | 1920 • 1200 |
| | | 10,8 Zoll: 250 • 175 • 70 | 1920 • 1280 |
| e) | Tablet | 7 Zoll: 188 • 107 • 11 | 1024 • 600 |
| | | 8 Zoll: 215 • 123 • 8,6 | 1280 • 800 |
| | | 8,9 Zoll: 240 • 135 • 10 | 1280 • 800 |
| | | 10 Zoll: 254 • 155 • 8 | 1920 • 1200 |

*: Grundfläche, zugeklappt.

Tabelle 6.5 Kleine Personalcomputer. Abmessungen und Bildschirmauflösungen.

| Merkmal | Netbook | Mini-PC, PC-Stick | TV-Box | Tablet |
|------------------------------------|----------------------|-------------------|----------------------|---|
| Bildschirm | Ja | – | Ja | Ja |
| Tastatur | Ja | – | – | – |
| Maus | – | – | – | – |
| Touchpad | Ja | – | – | – |
| Touchscreen | Manche Modelle | – | Ja | Ja |
| Netzbetrieb | Ja | Ja | Ja | – |
| Mehrere Anschlüsse (USB, Netzwerk) | Ja | Ja | Ja | – |
| Mechanischer Aufbau | Kompakt (All in One) | Einzelne Geräte | Kompakt (All in One) | Kompakt und flach. Autonom betriebsfähig. Ergänzungen außerhalb |
| Eigenbaulösungen* | Nicht erforderlich | Ja | Nicht erforderlich | Ja |

*: Wenn es nach etwas aussehen soll. Man kann natürlich die Teile auch einfach auf den Tisch legen...

Tabelle 6.6 Kleine Personalcomputer. Ausstattungsmerkmale. "–": nicht eingebaut. Müsste bei Bedarf hinzugefügt werden.

| Schnittstelle | Anschluss über | Anmerkungen |
|---------------|--------------------------|---|
| USB | Wandler USB auf seriell | Die einfachste Lösung. Kabelanschluss. Höhere Datenraten möglich. Latenzzeiten vergleichsweise kurz. Es kann sein, dass bei USB-Nutzung nur Akkubetrieb möglich ist*. |
| Bluetooth | Bluetooth Serial Adapter | Drahtlose Übertragung. Latenzzeiten praktisch nicht vorhersagbar. Bei Verzicht auf USB-Nutzung Netzbetrieb (parallel zum Laden) möglich*. |
| WLAN | WiFi Device Server | |

*: Das betrifft Tablets mit nur einer USB-Buchse Micro-AB (USB 2.0 OTG). Mit USB Type-C gibt es das Problem nicht.

Tabelle 6.7 Tablets mit Modulen verbinden.

| Der Kennungskontakt im USB-Kabelstecker | Tablet ausgeschaltet | Tablet eingeschaltet |
|---|-----------------------------|--|
| ID = 0 = Hostbetrieb | Tablet kann geladen werden* | Tablet arbeitet als Host und liefert 5 V über den USB. Akkubetrieb. Kein Laden |
| ID = 1 (z. B. Ladekabel) | Tablet kann geladen werden | Tablet am USB inaktiv. Kann geladen werden. Wenn Ladespannung anliegt, ist es Netzbetrieb mit parallelem Laden |

*: Das ist nicht immer sauber implementiert. Manche Tablets lassen sich über ein OTG-Kabel (Micro-A) laden, wenn sie ausgeschaltet sind, andere nicht. Der OTG-Stecker muss dann gezogen und ein echtes Ladekabel mit Micro-B-Stecker muss direkt angesteckt werden².

Tabelle 6.8 Betriebsspannungsversorgung über die USB-Buchse Micro-AB.

-
- 2: Die ehrgeizigere Selbstbaulösung wäre eine selbst konfektioniertes Kabel mit umschaltbaren ID-Pin...

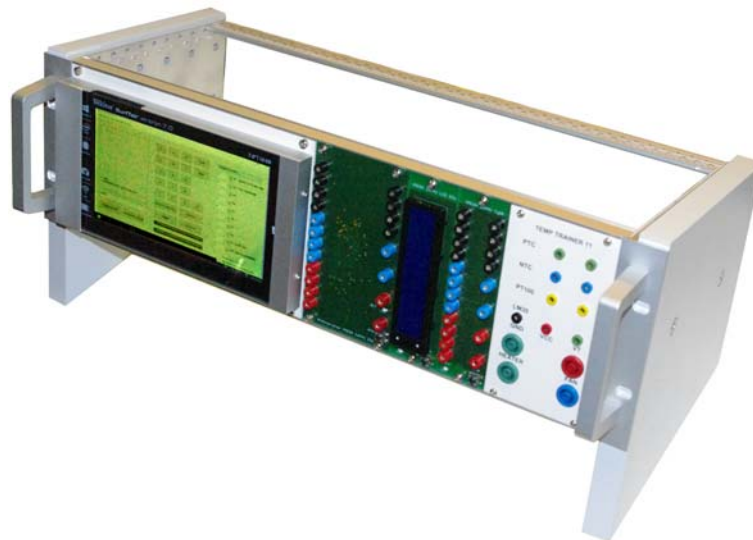


Abb. 6.15 Ein Tablet 7" vor einer Frontplatte 3 HE, 42 TE. Der Computer belegt einen halben Baugruppenträger. Man kann also auch zwei Computer nebeneinandersetzen.

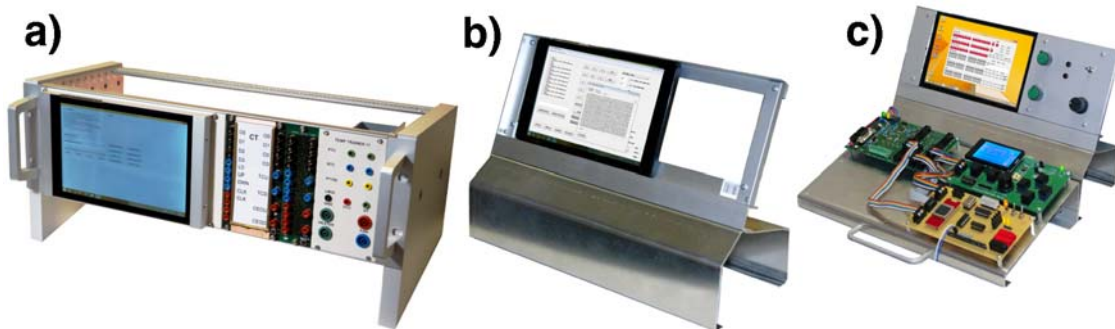


Abb. 6.16 So kann ein Tablet 8" adaptiert werden. a) vor einer Frontplatte 3HE, 49 TE, b) und c) in einem selbst gebauten Rahmen mit Sockel..



Abb. 6.17 Ein erster Versuch: zwei Inkrementalgeber, zwei Zweifarben-LEDs, ein Kippschalter mit drei Stellungen, ein Drehschalter mit fünf Stellungen.



Abb. 6.18 Dem PC gegenüber so tun, als wäre eine USB-Tastatur angeschlossen. Hier im Experiment. Der Schaltkreis (Pfeil) kann ein angeliefertes Bitmuster in einen Tastaturcode umsetzen und an den PC liefern. Das Bitmuster wird von einem Mikrocontroller erzeugt.

| | |
|--------------------|-------------|
| 640 • 480 (VGA) | 1366 • 768 |
| 800 • 400 | 1280 • 800 |
| 800 • 480 (WVGA) | 1920 • 1080 |
| 1024 • 600 (WSVGA) | 1920 • 1200 |
| 1024 • 768 | 1920 • 1280 |

Tabelle 6.9 Typische Bildschirmauflösungen kleiner PCs und Monitore.

| Formfaktor | Auflösung (Pixel) | Abmessungen (mm) |
|------------|-------------------|------------------|
| 7 Zoll | 800 • 480 | 16:9: 155 • 87 |
| | 800 • 400 | 16:10: 150 • 94 |
| | 1024 • 600 | |
| 8 Zoll | 1280 • 800 | 16:9: 177 • 99 |
| | | 16:10: 173 • 107 |
| 8,9 Zoll | 1920 • 1200 | 16:10: 191 • 119 |
| 10,1 Zoll | 1024 • 600 | 16:9: 223 • 125 |
| | 1280 • 800 | 16:10: 217 • 135 |
| 10,8" | 1920 • 1280 | 16:10: 232 • 145 |
| 11,6" | 1920 • 1200 | 16:10: 249 • 155 |

Tabelle 6.10 Typische Bildschirmabmessungen. Richtwerte. Rechnerisch ermittelt und abgerundet.

| Seitenverhältnis | Der Wurzelausdruck | Rechnung (Diagonale in Zoll) |
|------------------|--------------------|---|
| 16:9 = 1,78 | 2,04 | Senkrecht: Diagonale • 12,45 Waagrecht: Senkrecht • 1,78 |
| 16:10 = 1,6 | 1,89 | Senkrecht: Diagonale • 13,44 Waagrecht: Senkrecht • 1,6 |

Tabelle 6.11 Bildschirmabmessungen berechnen.

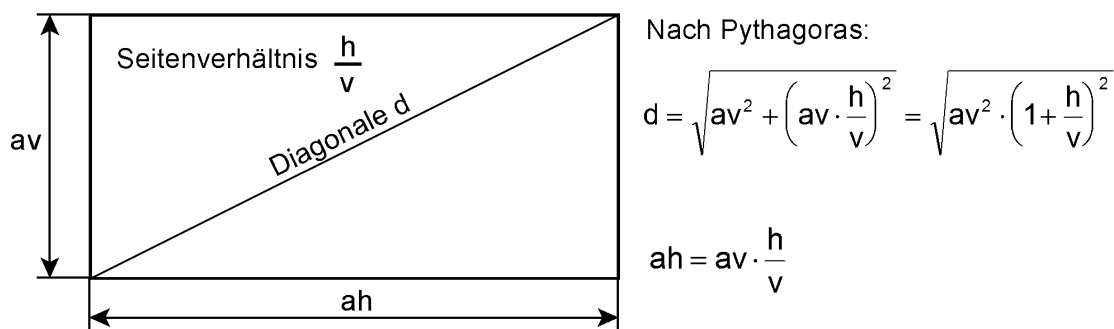


Abb. 6.19 Bildschirmabmessungen.

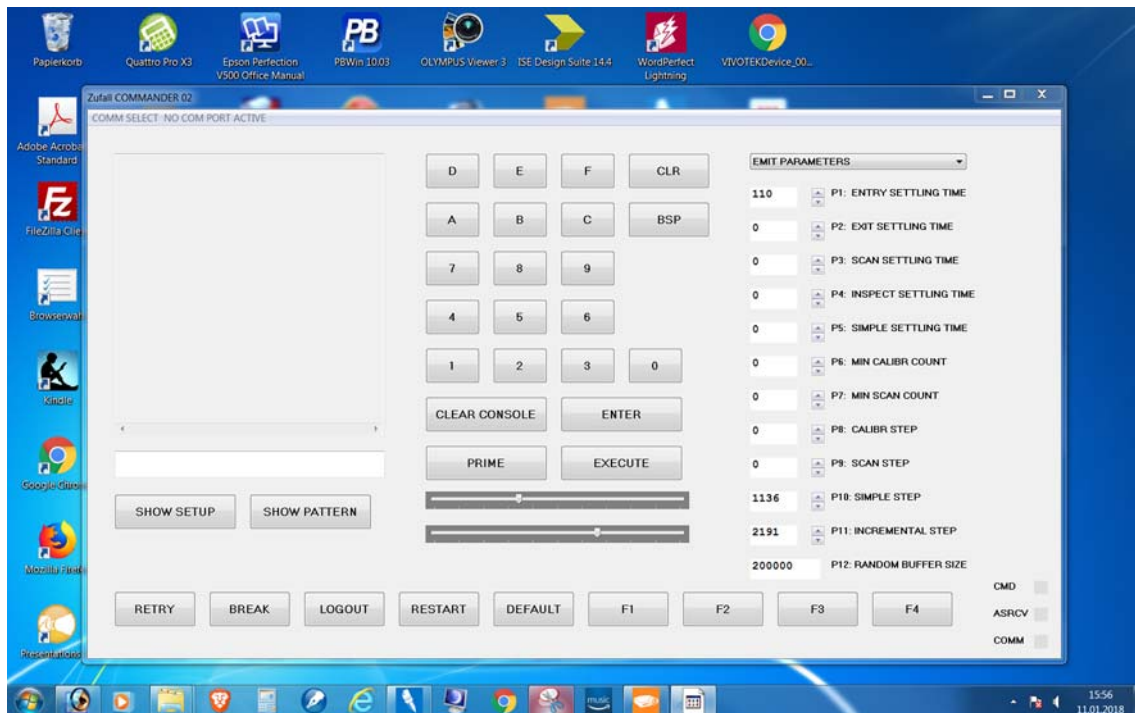


Abb. 6.20 Ein anwendungsspezifisches Bedienfenster. So erscheint es auf einem großen Bildschirm.

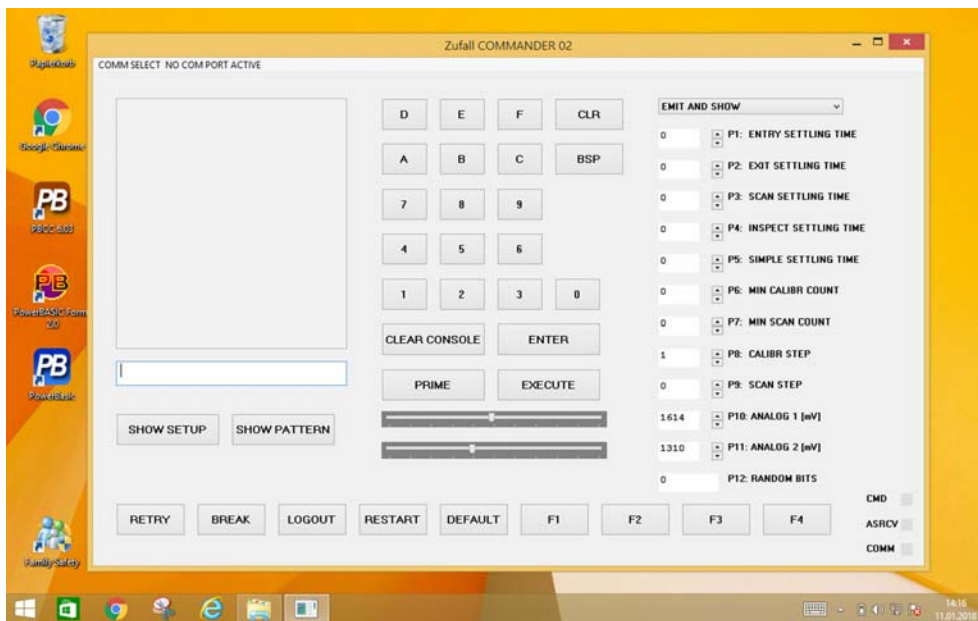


Abb. 6.21 Das gleiche Programm läuft hier auf einem 8"-Tablet. Auf dem Bildschirm ist alles gut zu erkennen. Manche Steuerelemente sind aber praktisch nicht mehr zu bedienen.

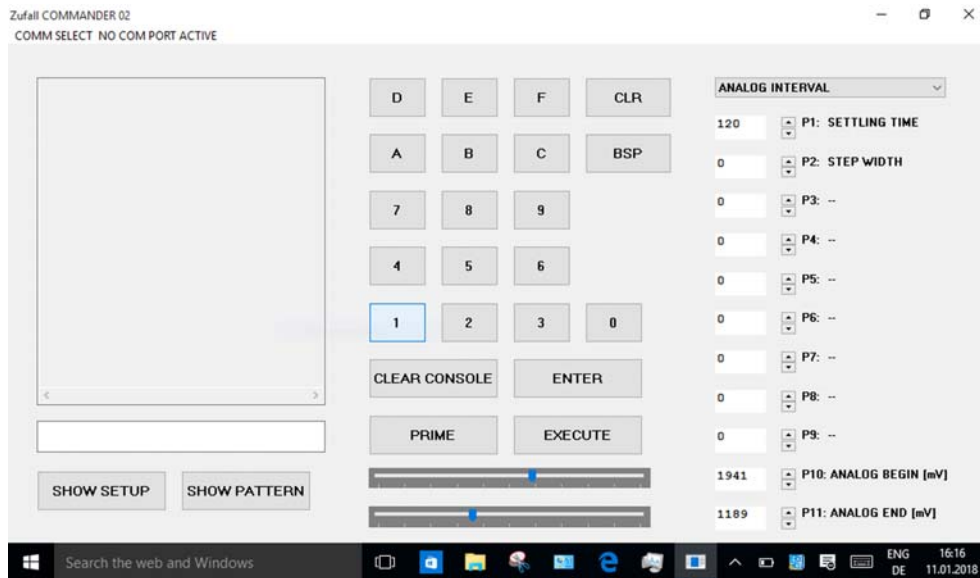


Abb. 6.22 Auf einem 7"-Tablet fehlen die Steuerelemente am unteren Rand. Immerhin ist die Schrift auf dem Bildschirm noch gut lesbar, die Steuerelemente sind noch gut zu erkennen. Damit alles draufpasst, müsste man im Programm etwas nachbessern.

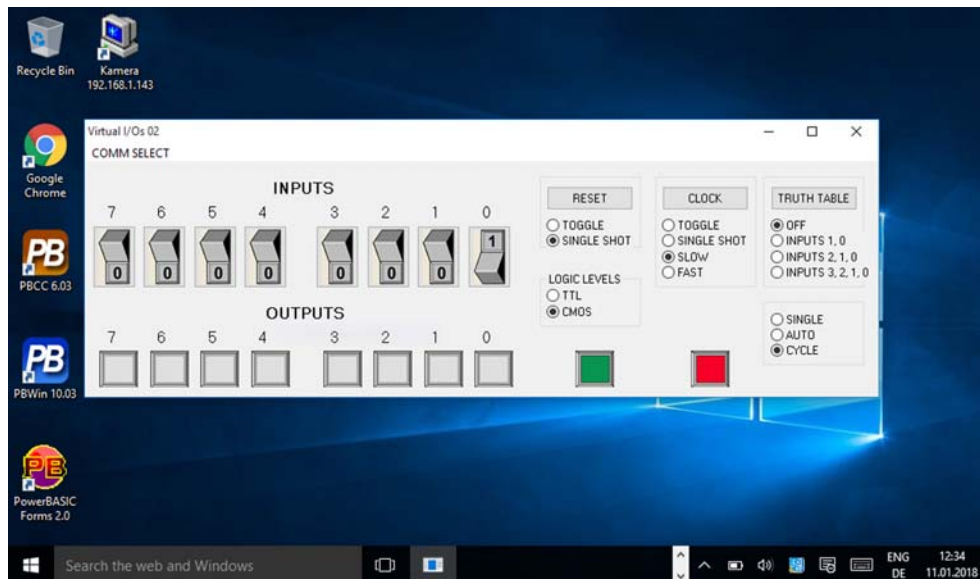


Abb. 6.23 Eine Bedientafel zum Experimentieren mit Digitalschaltungen, dargestellt auf einem 7"-Tablet.

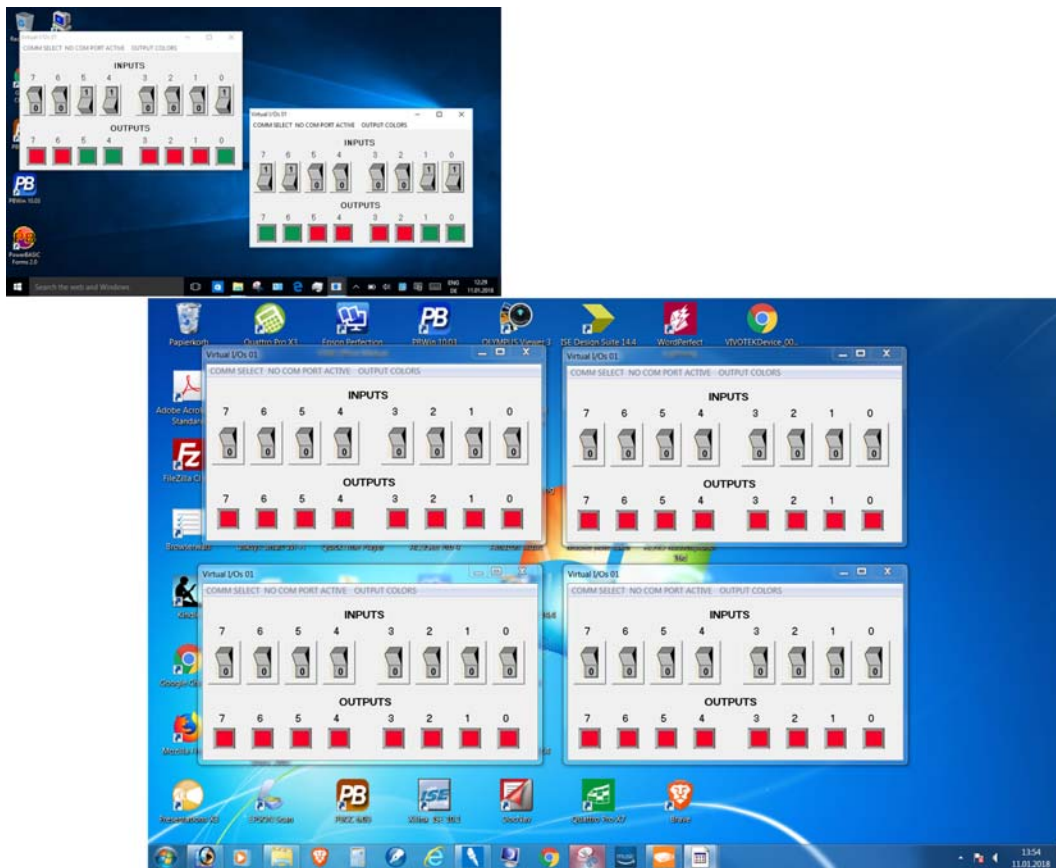


Abb. 6.24 Was auf dem 7"-Tablet die richtige Größe hat, ist auf dem 22"-Bildschirm viel zu groß (Abbildung unmaßstäblich).

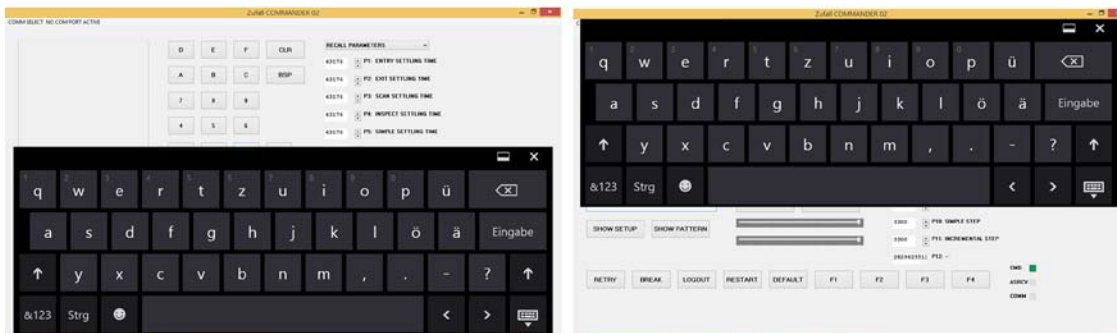


Abb. 6.25 Die Bildschirmtastatur des Systems verdeckt die Eingabefelder – ganz gleich, wie man es anstellt...

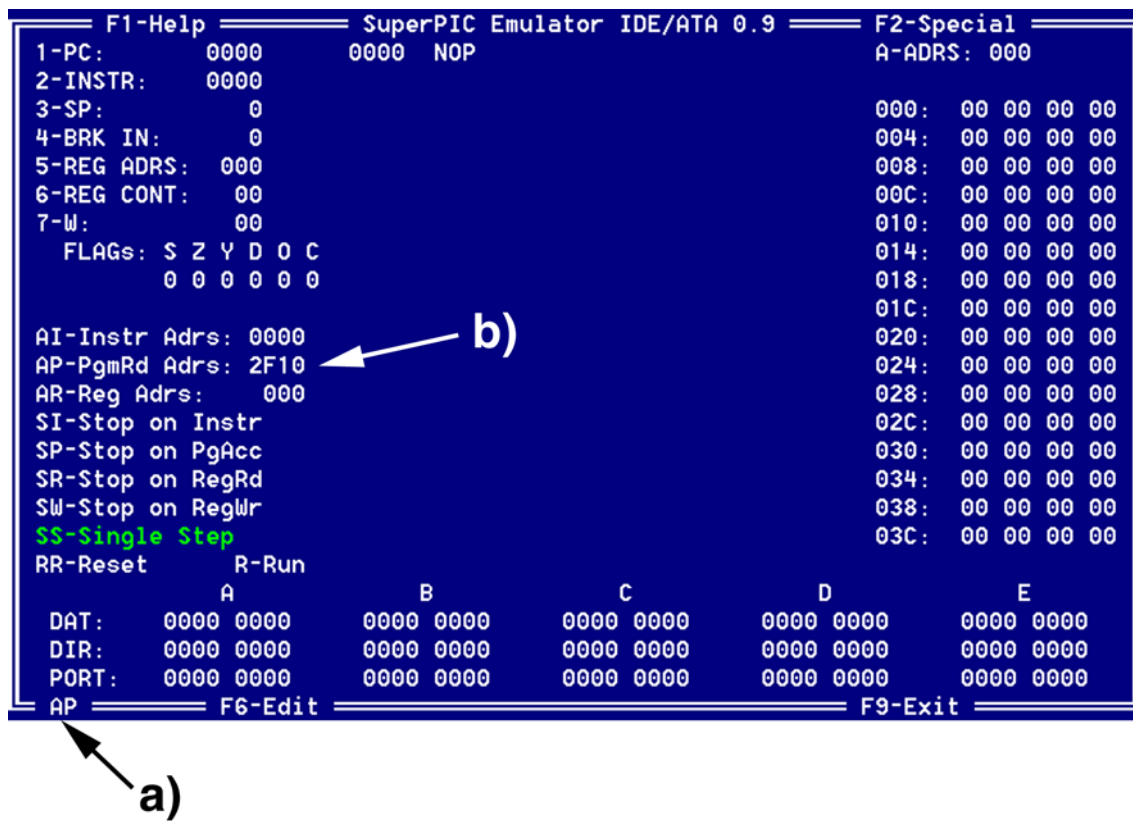


Abb. 6.26 Die Bedienoberfläche eines Emulators. Das Bild enthält eine Übersicht über die Bedienfunktionen und einige Eingabefelder. Funktionsauswahl durch Tasteneingabe ins Eingabefeld a). Die hier gezeigte Eingabe "AP" wählt ein Feld b) zum Eingeben einer Vergleichsstoppadresse aus.

7. Mikrocontroller

7.1 Mikrocontrollerfamilien

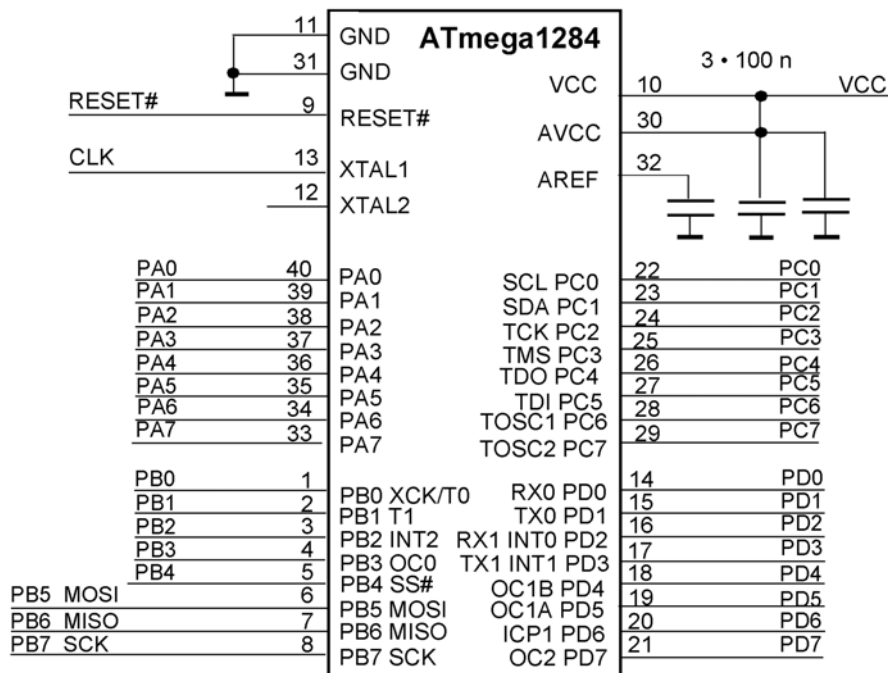


Abb. 7.1 Ein ATmega-Mikrocontroller im DIL-Gehäuse mit 40 Pins. Der ATmega 1284 hat eine besonders reichhaltige Ausstattung. Für kleinere Projekte genügt oftmals der ATmega16. Seine peripheren Funktionseinheiten sind in Assembler einfacher zu programmieren (es sind weniger, sie sind teils nicht so kompliziert, und sie sind alle über E-A-Adressen erreichbar).

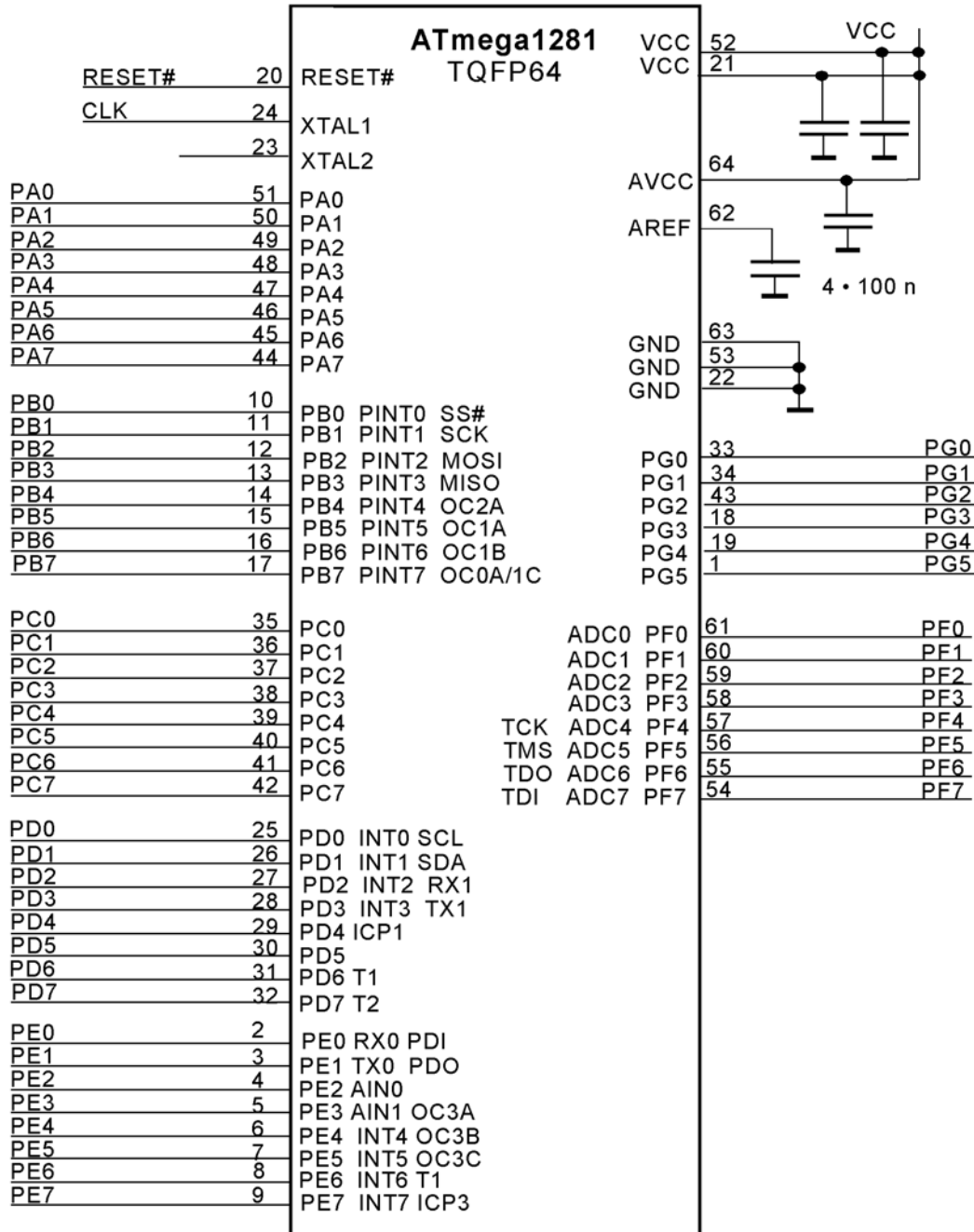


Abb. 7.2 Ein ATmega-Mikrocontroller in einem SMD-Gehäuse. Hier ein Flatpack mit 64 Anschlüssen.

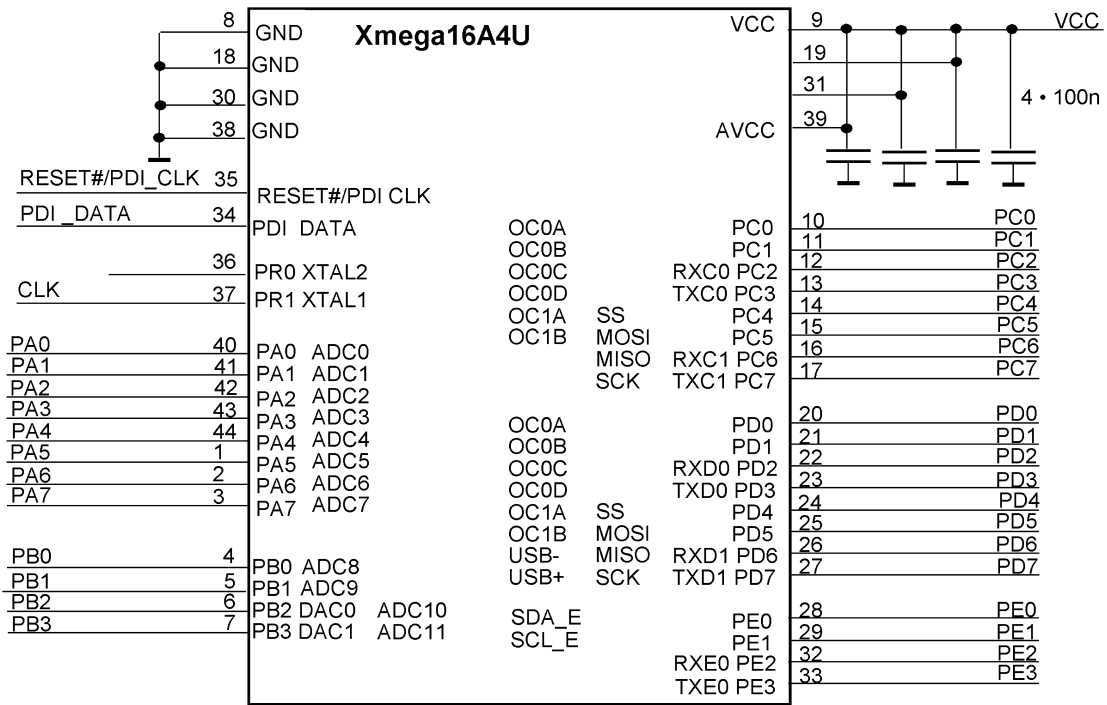


Abb. 7.3 Ein kleinerer Xmega-Mikrocontroller. Das Gehäuse ist ein Flatpack mit 44 Anschlüssen.

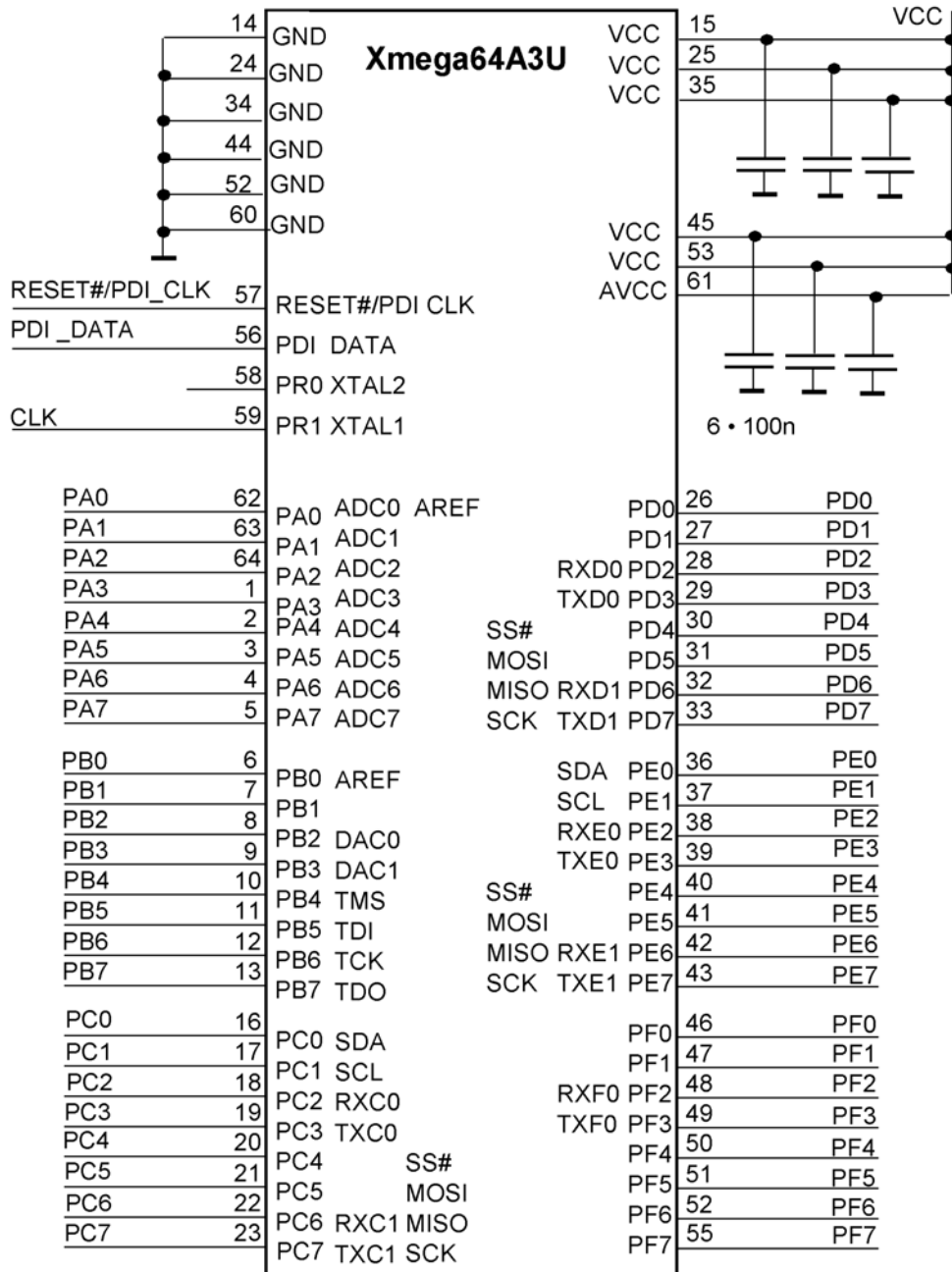


Abb. 7.4 Ein größerer Xmega-Mikrocontroller. Das Gehäuse ist ein Flatpack mit 64 Anschlüssen. Es lässt sich noch von Hand löten.

7.2 Takt

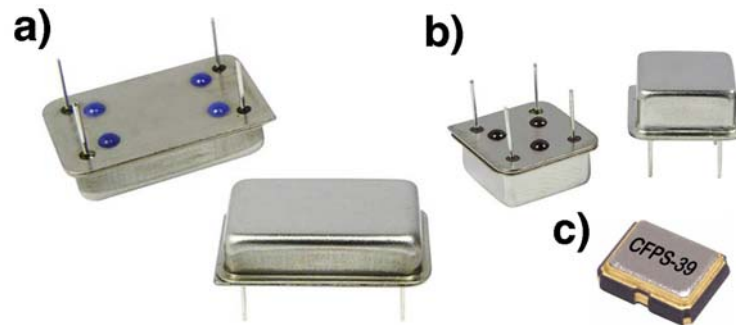


Abb. 7.5 Quarzoszillatoren. a) DIL 14polig, b) DIL 8polig, c) SMD.

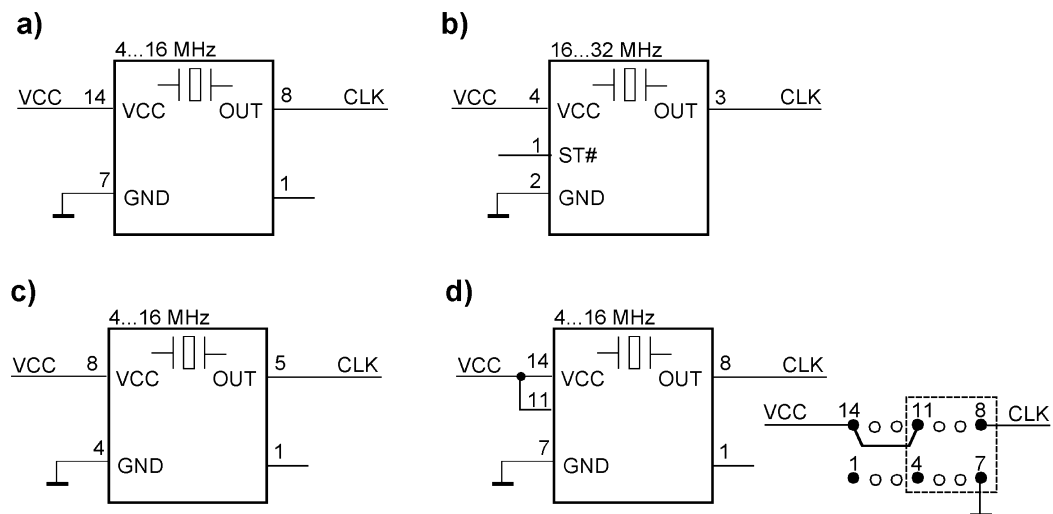


Abb. 7.6 Quarzoszillatoren in der Schaltung. a) DIL 14polig, b) SMD, c) DIL8polig, d) Kombinations-Layout für 14- und 8polige DIL-Gehäuse.

7.3 Rücksetzen

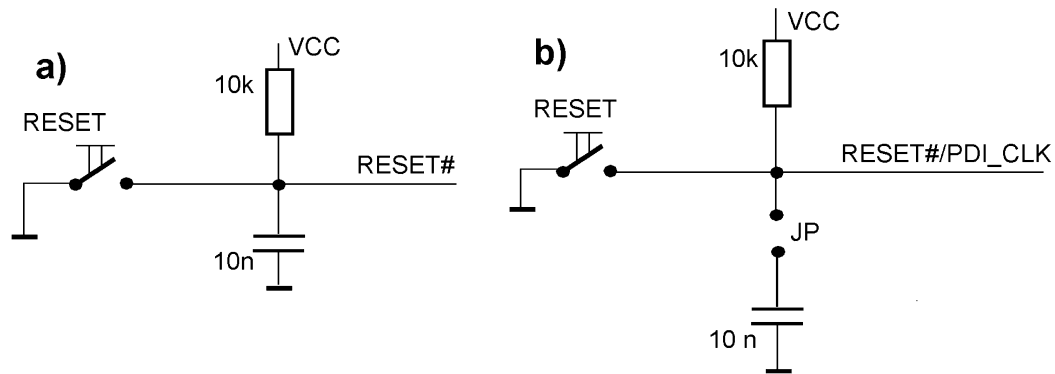


Abb. 7.7 Die Bildung des Rücksetzsignals. a) für ATmega, b) für Xmega.

7.4 Programme in den Mikrocontroller laden

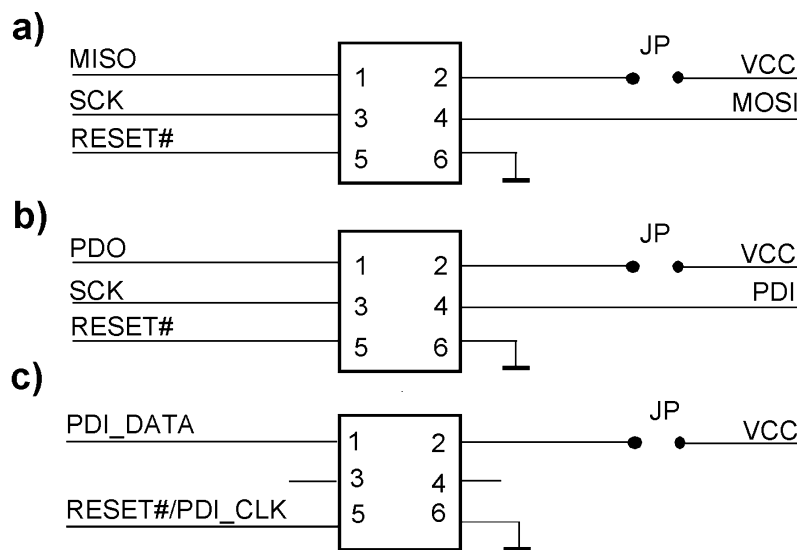


Abb. 7.8 Programmieranschlüsse. a) und b) ATmega, c) Xmega.

7.6 Programme zum Laufen bringen



Abb. 7.9 Geräte zum Debugging (eine kleine Auswahl). Sie sind an einen PC und an das Zielsystem anzuschließen. Zu jedem dieser Geräte gibt es ein Sortiment an Adaptern und Kabeln für die verschiedenen Debuggingschnittstellen (JEDEC, PDI, debugWire usw.). ICD = In Circuit Debugger, ICE = In Circuit Emulator.

7.7 Programme schreiben

7.8 Mehrprozessorsysteme

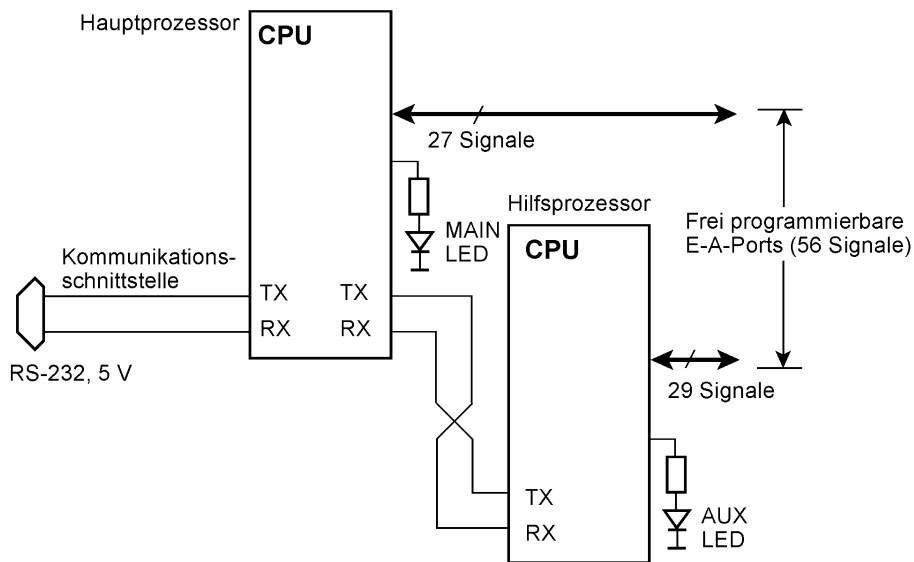


Abb. 7.10 Ein elementares Zweiprozessorsystem, das viele frei programmierbare E-A-Signale unterstützen kann. Von insgesamt 56 Signalen unterstützt der Hauptprozessor 27, der Hilfsprozessor 29.



Abb. 7.11 Ein Ausführungsbeispiel – das CPLD-Lehrgerät 12. Zwei Mikrocontroller, 56 frei programmierbare Signale. Vorzugsweise Nutzung als quasistatischer Tester oder Peripherienachbildung. Anschluss der zu testenden Einrichtungen über Wannienstifteleisten oder durch Aufstecken auf den PC/104-Steckverbinder in der Mitte der Platine.

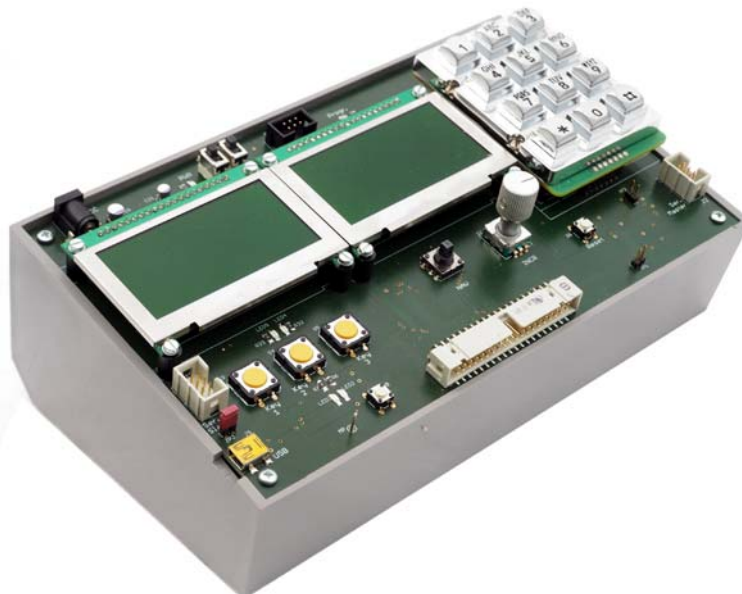


Abb. 7.12 Das All-in-One-Gerät Typ 2 hat zwei Prozessoren, einen für die Schnittstellen, Tasten und Anzeigen, einen für die universelle Ein- und Ausgabe. Hier ist vieles beieinander, was man ansonsten erst zusammenstöpseln müsste...

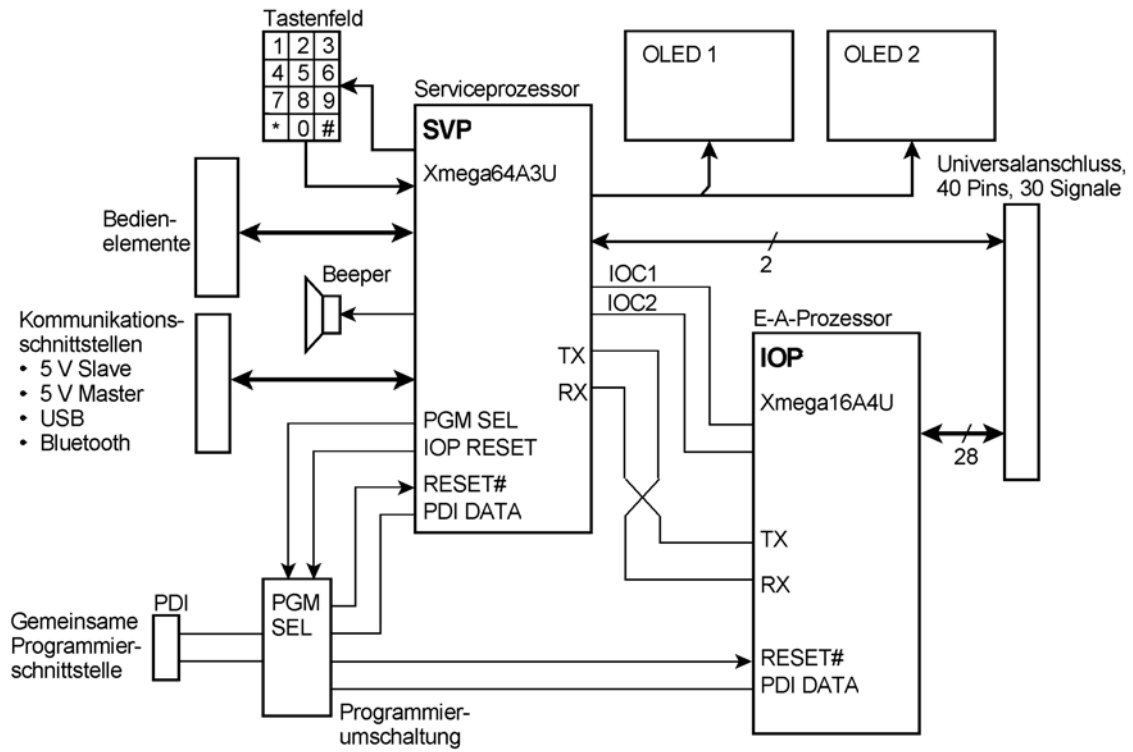


Abb. 7.13 Das All-in-One-Gerät Typ 2 im Überblick.

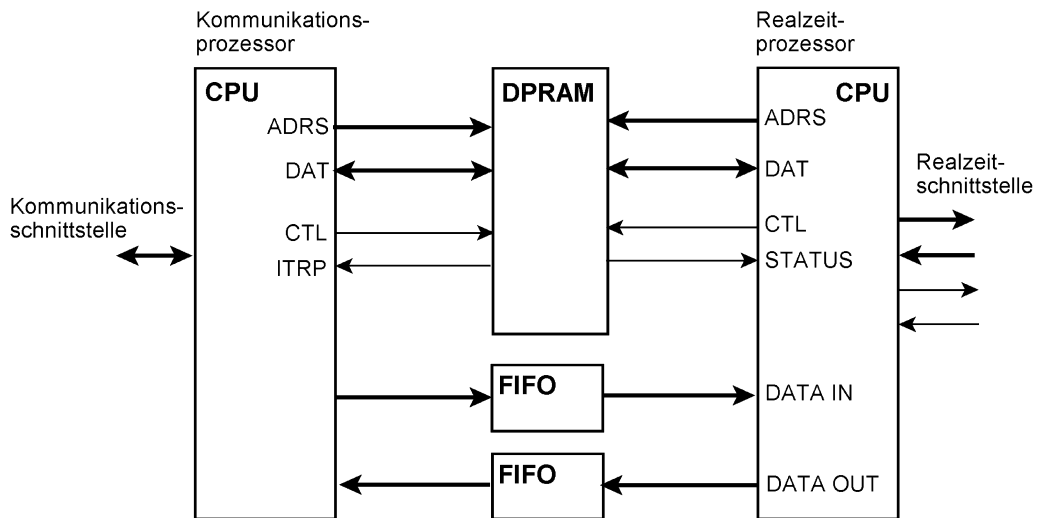


Abb. 7.14 Eine bewährte Grundsatzlösung der Zweiprozessorkopplung für harte Realzeitanforderungen.

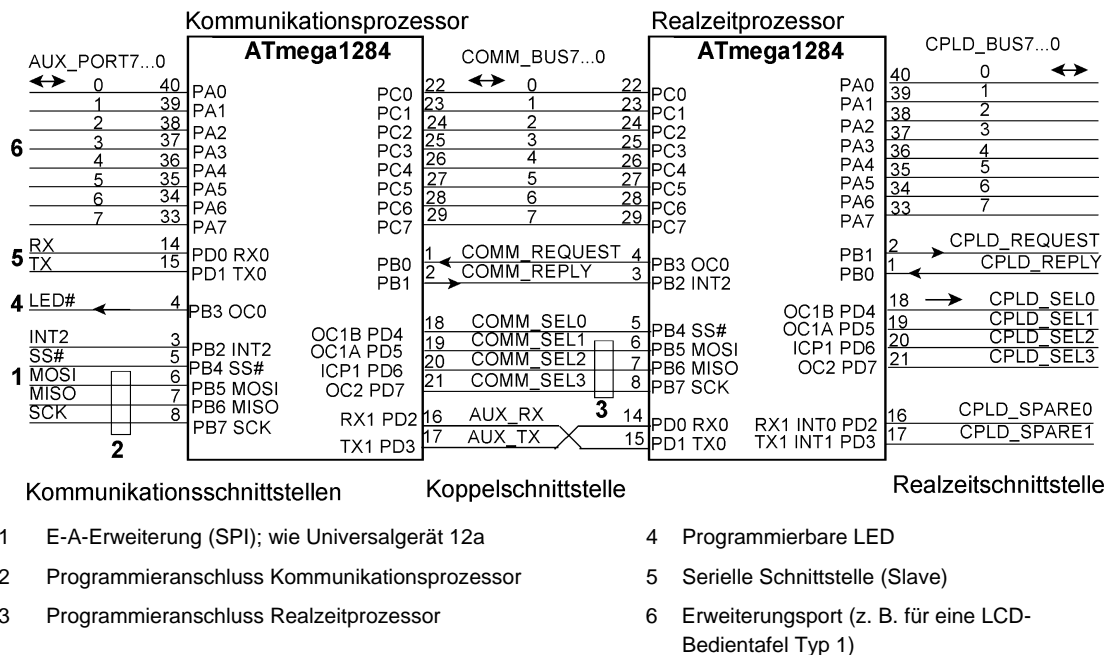


Abb. 7.15 Direkte Kopplung zweier Mikrocontroller für harte Realzeitanforderungen. Es gibt ein ergänzendes CPLD-Modul, das dem Realzeitprozessor nachgeschaltet werden kann. Die Signalbezeichner rechts außen beziehen sich auf diese Nutzungsweise.

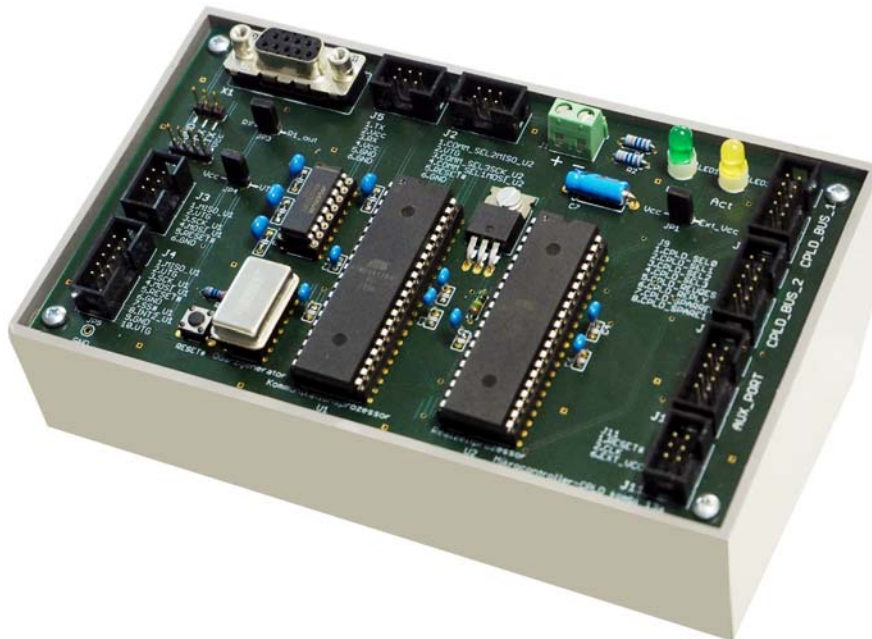


Abb. 7.16 Die Realzeitkombination 13 – ein Modul mit zwei gekoppelten Mikrocontrollern. Links der Kommunikationsprozessor, rechts der Realzeitprozessor.