

PCs als Mikrocontroller

Stand: 27. 10. 2003

Gliederung:

1. Einführung
2. Plattformen
3. Software
4. Schnittstellen
5. Schaltungstechnik
6. Anwendungspraxis

Inhalt

Der Einsatz von PCs als Mikrocontroller (für x-beliebige Steuerungsaufgaben). Wir verwenden einen PC anstelle eines Mikrocontrollers. Um die typischen Mikrocontroller-Ports nachzubilden, wird der PC mit Zusatzschaltungen erweitert.

Vorteile:

- preiswerter als ein gängiges Starterkit (das ja auch noch einen PC braucht (für die Entwicklungsumgebung)),
- leistungsfähiger,
- viel komfortabler (Bedienung, Anzeige, Dateiverwaltung),
- viel kürzere Turnaround-Zeiten (vom Programmieren bis zum lauffähigen Programm). Kein Download erforderlich.
- preisgünstige oder kostenlose Software,
- keine Maschinenabhängigkeit,
- E-A-Ports können leicht modifiziert werden. Konfiguration: PC - Entwicklungssoftware für Anwendung - CPLD-Entwicklungssoftware - Zusatzbeschaltung (in Form von CPLDs).

Herkömmlich:

PC - Starterkit - Anwendungsumgebung. Entwicklungsgang: Programmieren - Übersetzen - Download - Laufenlassen.

Die Alternative:

PC - Interfaceadapter - Anwendungsumgebung. Entwicklungsgang: Programmieren - Übersetzen - Laufenlassen. Keine Download-Zeiten, sondern praktisch sofortige Programmausführung. Genug Speicher, genug Leistung, Bedien- und Anzeigemittel des PCs zur Fehlersuchunterstützung einsetzbar.

Anwendungen, Zielgruppen:

- Einstieg in die Mikrocontroller-Programmierung,
- Selbstbauvorhaben,
- Entwicklungsvorhaben (Anwendungslösung wird auf PC entwickelt und auf den “echten” Mikrocontroller portiert),
- modulare Systemlösungen, z. B. in der Meß- und Prüftechnik.

Welche PCs?

- ältere Modelle (Weiternutzung),
- moderne PC-Baugruppen, vor allem die kleinen Motherboards, PC-104-Moduln usw.

Einige Stichpunkte zum Inhalt:

Einführung

Plattformen

- PC-Generationen
- Formfaktoren
- ältere PCs
- kleine kommerzielle Formfaktoren
- industrielle Formfaktoren
- Mechanik
- Stromversorgung

Software

- Was gibt es im Bereich der freien Software?
- Wie kann man unter Windows oder Linux arbeiten?
- Systemsoftware (DOS - Windows - Linux - EFI)
- Entwicklungssoftware (Assembler - C - Pascal - Basic - moderne Entwicklungsumgebungen)
- Mikrocontroller-Emulation
- Anwendungssoftware

Schnittstellen

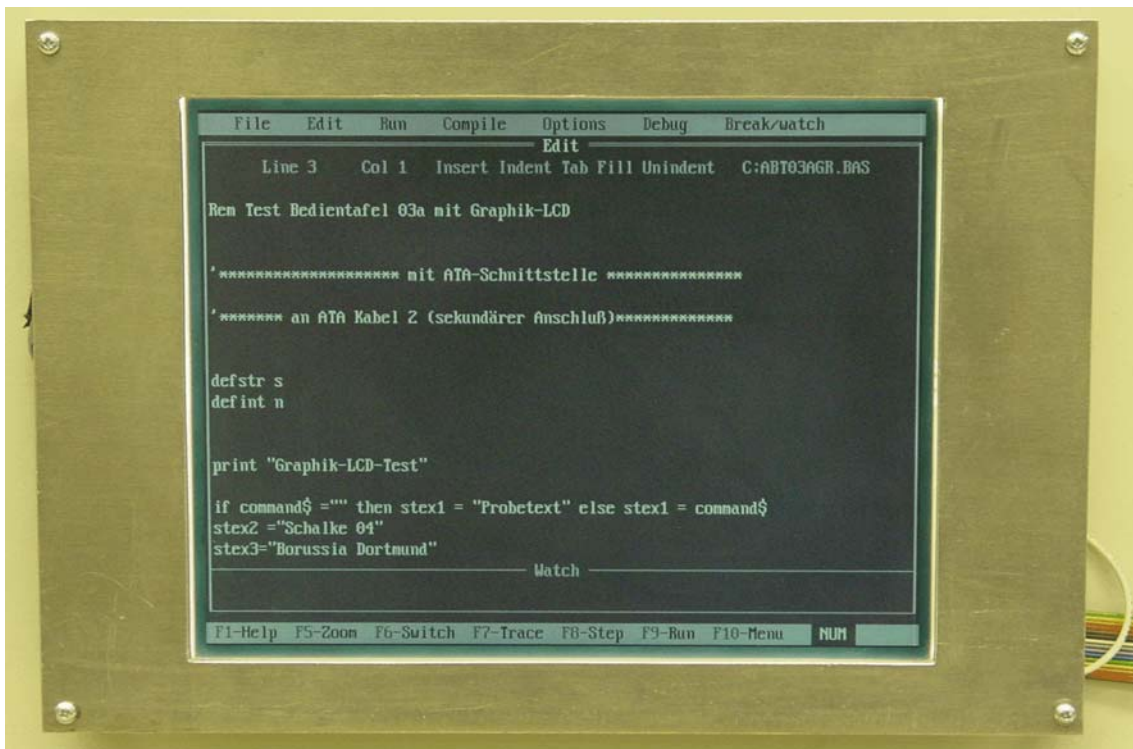
- ISA-Bus
- seriell
- parallel
- IDE/ATA
- Tastatur
- USB
- Netzwerk
- bitseriell, einfach (I2C, JTAG usw.)
- IEEE 488?

Schaltungstechnik

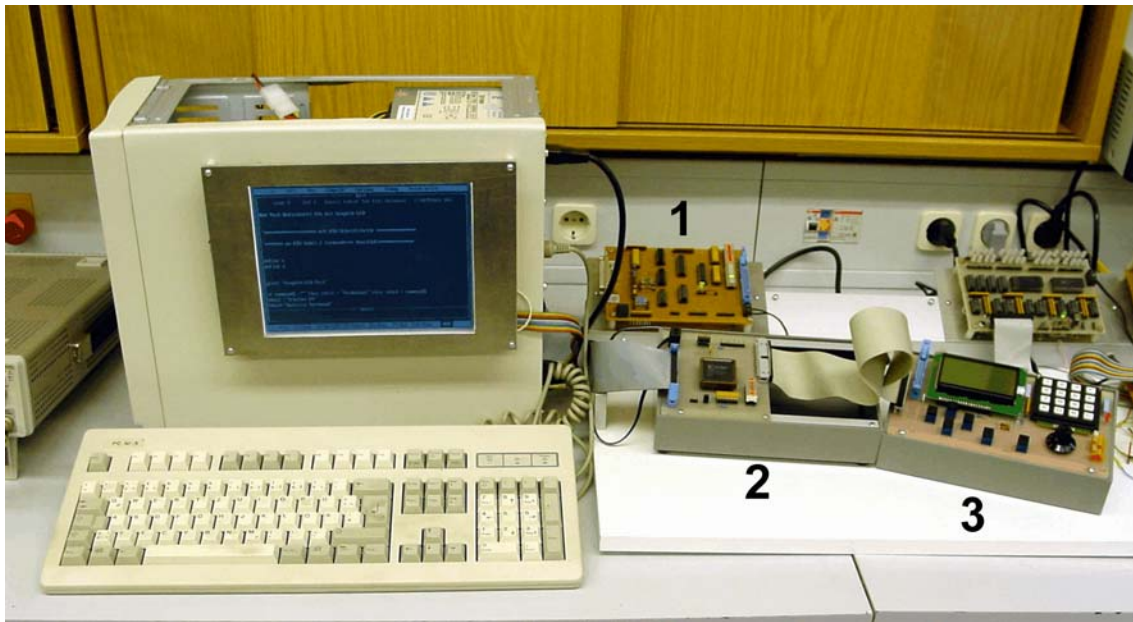
- E-A-Ports
- Sonderschaltungen (Zeitgeber, Impulsmustergeneratoren, Wandler usw.)
- extern angeschlossene Mikrocontroller
- Praxisprobleme: Hot Plugging, Partial Power Down, Metastabilität, Pegelwandlung
- Fehlersuchhilfen: Zustandsanzeigen, Ablaufspeicher, Stimuluserzeugung

Anwendungspraxis

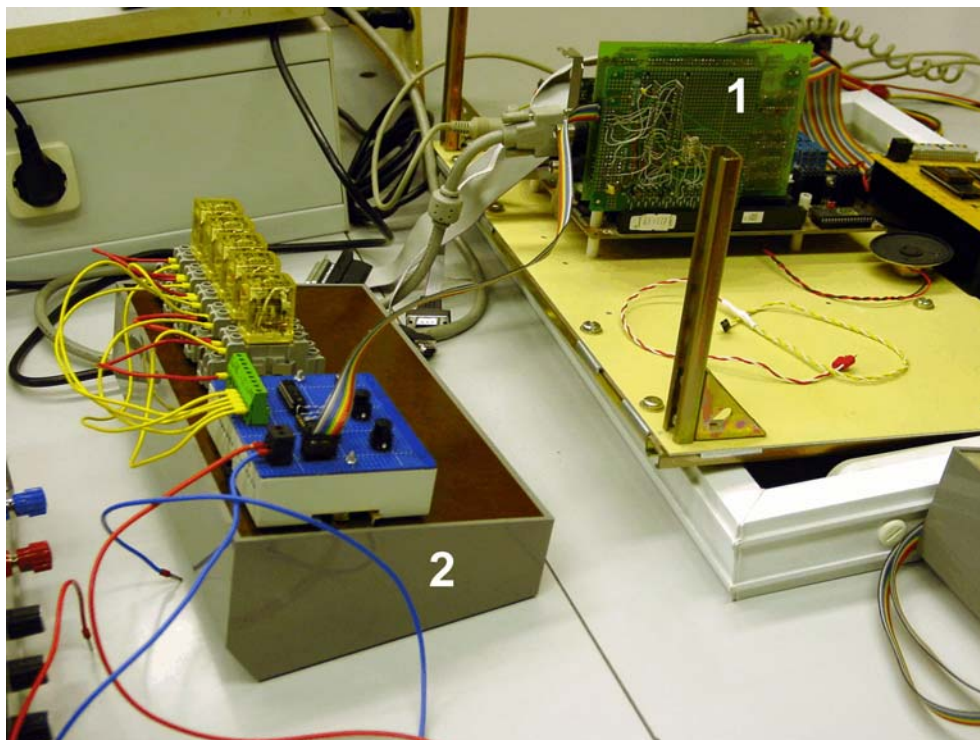
Bildbeispiele:



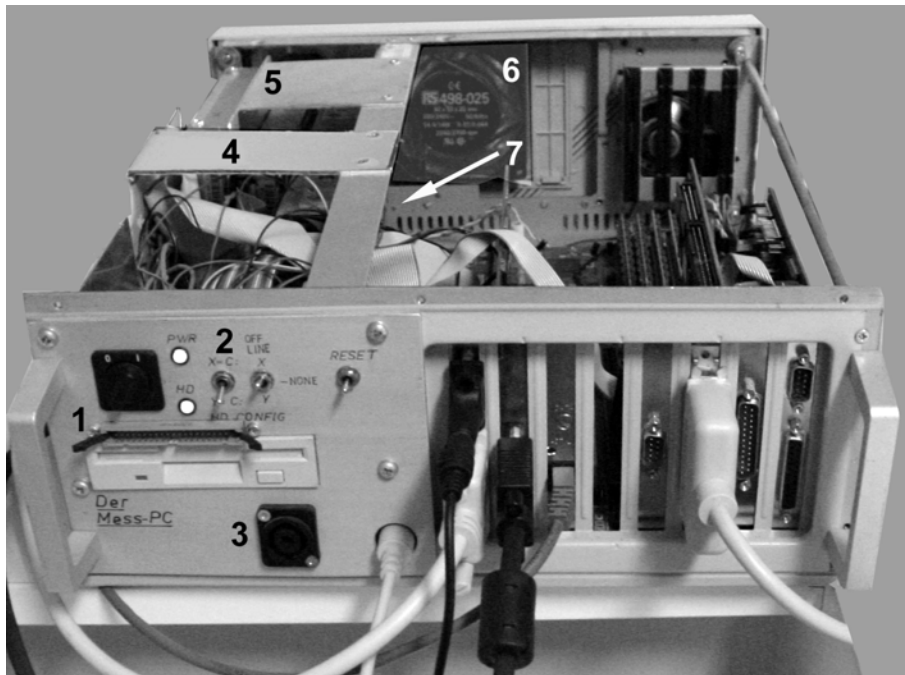
Ein älterer PC, als Mikrocontroller-Ersatz umgebaut (mit angeflanschem LCD-Bildschirm)



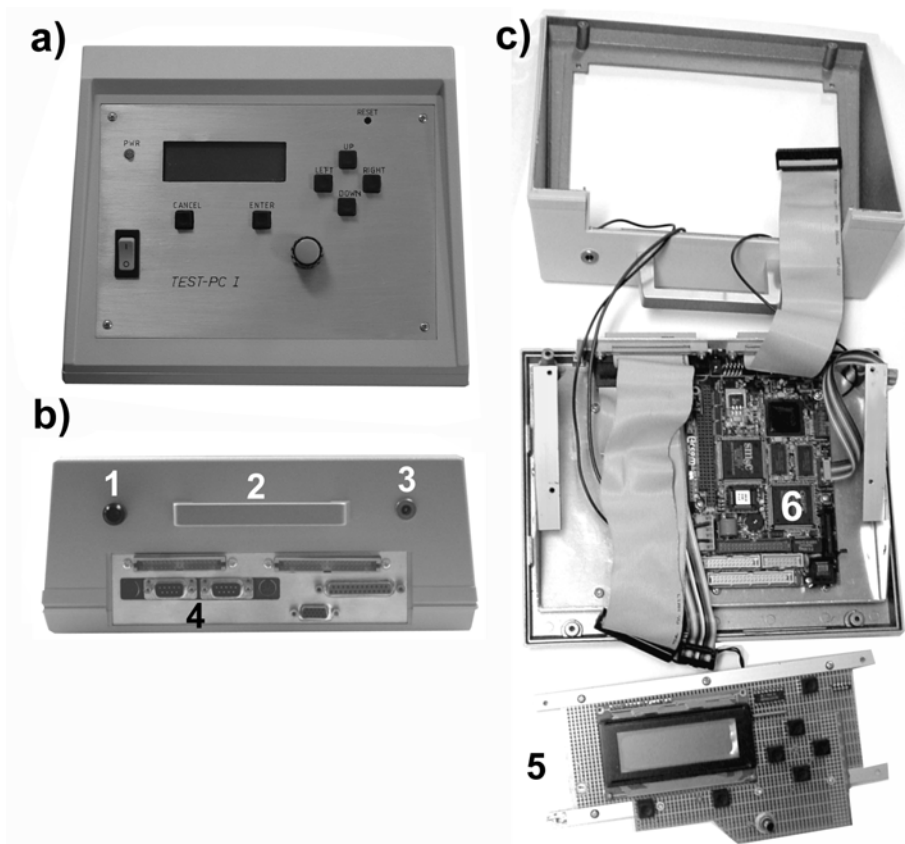
Der PC im Einsatz. 1 - Adapter an Parallelport; 2 - Adapter an IDE-Schnittstelle (5 E-A-Ports zu 8 Bits); 3 - Übungstafel mit LCD-Display und Tastenmatrix



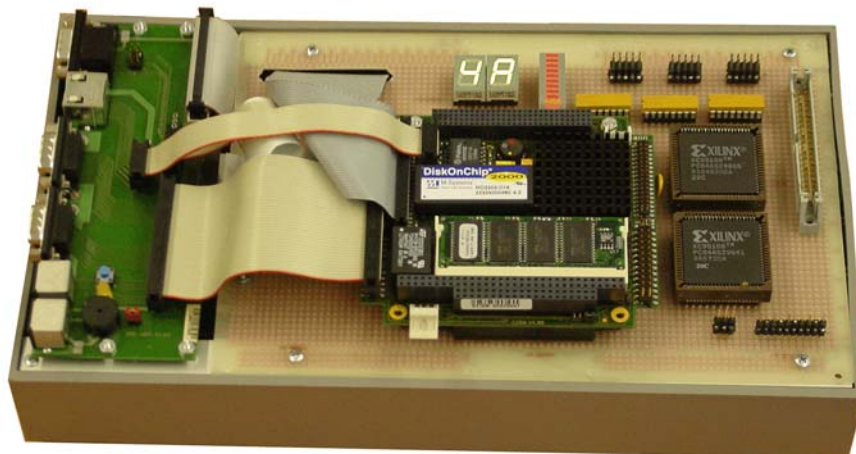
Ein Motherboard auf Prüftafel. Als Mikrocontroller eingesetzt. 1 - Portadapter am ISA-Bus; 2 - Übungsgerät zur Ansteuerung von Relais



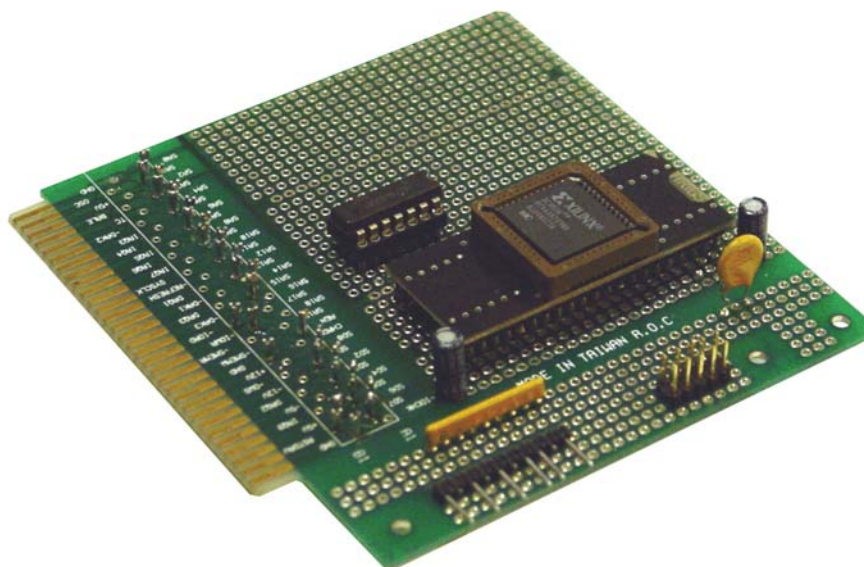
Ein umgebauter PC. Alle Slots vorn



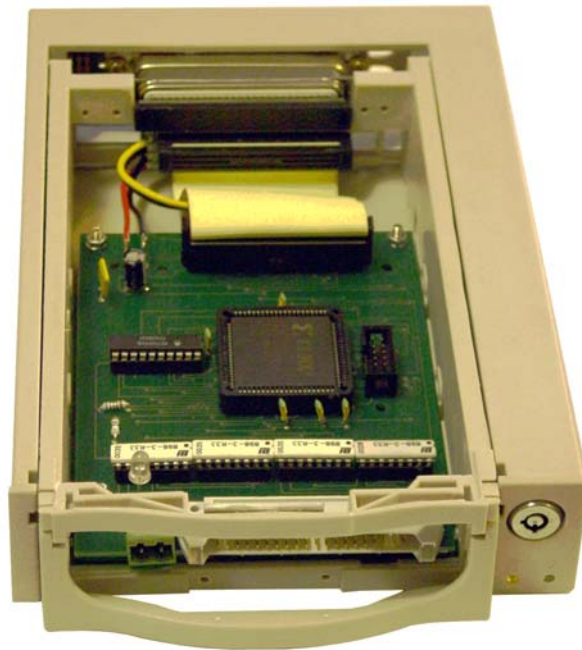
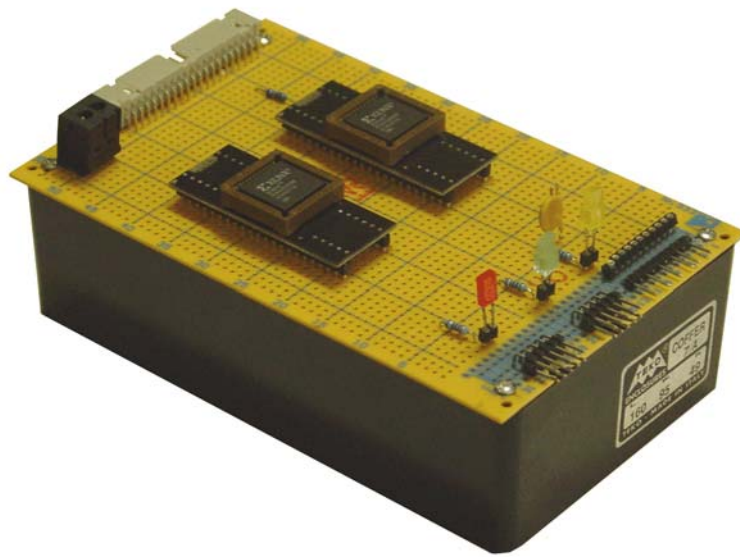
Selbstgebafter Computer auf Grundlage eines Industrie-PC-Moduls



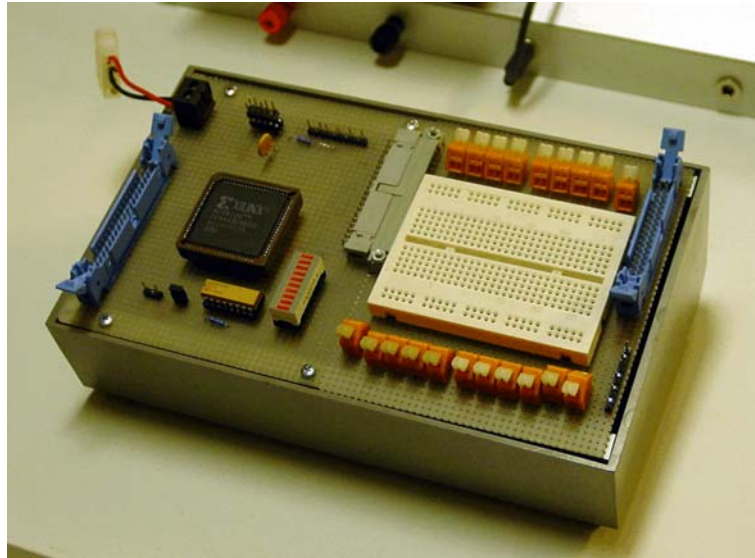
Ein weiteres Selbstbauprojekt mit Industrie-PC-Modul



Portadapter für ISA-Bus



Portadapter für IDE-Schnittstelle. Oben 2, darunter 4 8-Bit-Ports.



Portadapter für IDE-Schnittstelle (mit Übungstafel). 5 8-Bit-Ports