

Eingebettete Systeme

Prof. Dr. rer. nat. Peter Gerwinski

22. Dezember 2020

Eingebettete Systeme

<https://gitlab.cvh-server.de/pgerwinski/es>

1 Einführung

2 Einführung in Unix

3 TCP/IP in der Praxis

3.8 X11 und VNC

3.9 Programmierung

4 Versionsverwaltungssysteme

4.1 Historisches

4.2 diff und patch

4.3 Git

5 Bus-Systeme

5.1 Was sind Bus-Systeme?

5.2 RS-232

5.3 I²C (TWI)

5.4 SPI

...

...

3 TCP/IP in der Praxis

3.8 X11 und VNC

- Grafik-Bildschirm und Eingabegeräte über's Netz
- `DISPLAY`-Variable: X-Server: Rechner und Bildschirm
- `ssh -X`: X11-Forwarding

- VNC = Virtual Network Computing
- VNC-Server stellt Bildschirminhalt zur Verfügung
 - entweder: eigener, virtueller X11-Server
 - oder: ruft Inhalt von anderem (X11-) Bildschirm ab
- VNC-Client ruft Bildschirminhalt ab und stellt ihn dar
 - z. B. per X11
 - z. B. per Web-Interface: noVNC

3 TCP/IP in der Praxis

3.9 Programmierung

- Shell-Skripte: `nc` (`traditional` oder `OpenBSD`)
- C: Sockets, `select()`
- C++: Callbacks

Aktuelles Beispiel: Programmierung eines VNC-Servers per Web-Interface

- Grundlagen: RFC 6143
- Praxis: Untersuchung mit Wireshark
- Implementation: JavaScript, WebSockets, `websocketify`, Callbacks

4 Versionsverwaltungssysteme

4.1 Historisches

Bekannte Versionsverwaltungswerkzeuge:

1974 diff

1982 RCS – Revision Control System

1985 patch

1990 CVS – Concurrent Versions System

2000 SVN – Subversion

2005 Git

2005 Mercurial

... und *vielen* weiteren!

4 Versionsverwaltungssysteme

4.2 diff und patch

- Standard-Werkzeuge unter Unix
- `diff`: Textdateien miteinander vergleichen
- `diff -w` – Leerzeichen und Tabulatoren (whitespace) ignorieren
- `diff -c` – Kontext (context) anzeigen
- `diff -u` – Gemeinsamen (unified) Kontext anzeigen
- `diff -p` – Funktion (procedure) anzeigen
- `patch`: In `diff`-Datei gespeicherte Änderungen automatisch ausführen
- Dadurch möglich: parallele Änderungen an einer Datei (branches) automatisch zusammenführen (merge)
- Bei Konflikten entsteht eine neue `diff`-Datei mit der Endung `rej` (reject).

4 Versionsverwaltungssysteme

4.3 Git

- Name: „git“ (engl.) = „Idiot“
- Initiator: Linus Torvalds
„I'm an egotistical bastard, and I name all my projects after myself. First ,Linux', now ,Git'.“
- **Repository** – „Behälter“, in dem alle Dateien liegen
– einschließlich sämtlicher historischer Versionsn
- bei Git: dezentral
- **Branches** – parallele Versionen desselben Repository
- bei Git: leistungsfähige Werkzeuge, um Branches wieder zusammenzuführen

4 Versionsverwaltungssysteme

4.3 Git

- *commit* – Änderung in das Repository übernehmen
- *stage* – für *commit* vorgemerkte Änderungen
- Unterverzeichnis *.git*: Konfiguration und Repository

Befehle:

- *git status* – Status anzeigen
Welche Dateien sind in der *stage*? Welche sind neu?
- *git add* – Änderungen vormerken
- *git commit* – Änderungen in Repository übernehmen
- *git checkout -- <Datei>* – Datei aus Repository zurückholen
- *git push* – Änderungen in anderes Repository übertragen
- *git clean* – temporäre Dateien löschen
- *git init* – Verzeichnis für Arbeit mit Git vorbereiten

5 Bus-Systeme

5.1 Was sind Bus-Systeme?

Ein Bus ist ein System zur Datenübertragung zwischen mehreren Teilnehmern über einen gemeinsamen Übertragungsweg.

[https://de.wikipedia.org/wiki/Bus_\(Datenverarbeitung\)](https://de.wikipedia.org/wiki/Bus_(Datenverarbeitung))

Beispiele:

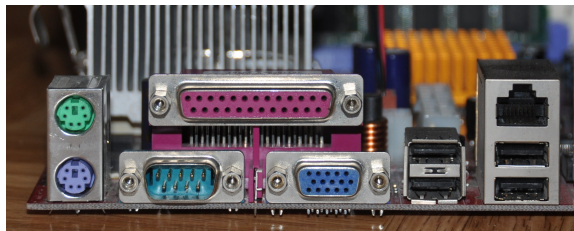
- Computer kommuniziert mit Peripherie
- Computer kommunizieren (direkt) miteinander
- Prozessor kommuniziert mit externem Speicher
- Teile eines Prozessors kommunizieren miteinander

5 Bus-Systeme

5.1 Was sind Bus-Systeme?

Standard-Computer:

- Einsteckkarten: PCI (und Vorgänger)
- Festplatten: SATA (und Vorgänger)
- USB, FireWire, ...
- Ethernet, CAN-Bus, ...
- WLAN, BlueTooth, IR, ...
- PS/2, RS-232, Centronics



5 Bus-Systeme

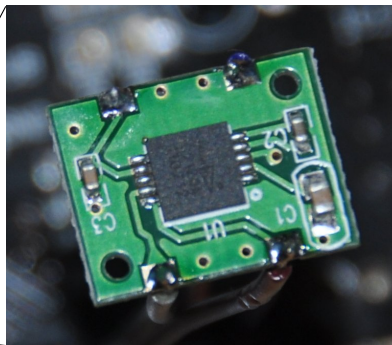
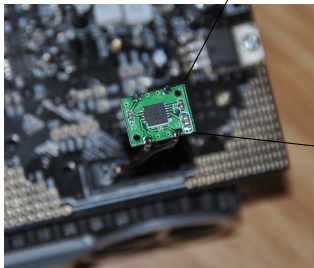
5.1 Was sind Bus-Systeme?

Standard-Computer:

- Einsteckkarten: PCI (und Vorgänger)
- Festplatten: SATA (und Vorgänger)
- USB, FireWire, ...
- Ethernet, CAN-Bus, ...
- WLAN, BlueTooth, IR, ...
- PS/2, RS-232, Centronics

Minimal-Hardware:

- RS-232
- I²C (TWI)
- SPI



5 Bus-Systeme

5.1 Was sind Bus-Systeme?

<i>seriell</i>	jedes Bit einzeln übertragen
<i>parallel</i>	mehrere Bits gleichzeitig
<i>synchron</i>	Abgleich über Steuerleitung: <i>Takt</i>
<i>asynchron</i>	Abgleich über Zeitvereinbarungen
<i>Punkt-zu-Punkt</i>	genau zwei Teilnehmer
<i>busfähig</i>	mehrere Teilnehmer, mit <i>Adressierung</i>

- I²C: seriell, synchron, mit Adressierung
- RS-232: seriell, asynchron, Punkt-zu-Punkt
- RS-485, USB, CAN: seriell, asynchron, mit Adressierung
- SPI: seriell, synchron, Punkt-zu-Punkt oder mit Adressierung

5 Bus-Systeme

5.2 RS-232

seriell

- *TX*: 1 Leitung für Daten
- *RX*: ggf. 1 Leitung für Daten in der anderen Richtung
- *GND*: gemeinsame *Masse*
- evtl. zusätzliche Steuerleitungen

asynchron

- *keine* Taktleitung für Abgleich, wann Daten anliegen
- Stattdessen: Abgleich über Zeitvereinbarungen

→ Jeder Teilnehmer braucht eine eigene Zeitbasis.

Punkt-zu-Punkt

- nur 2 Teilnehmer vorgesehen

5.2 RS-232

Synchronisation

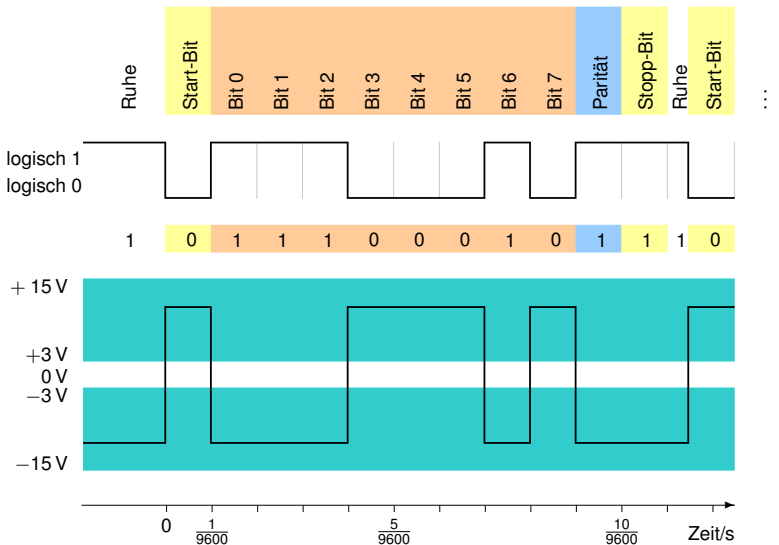
Daten

Check

9600 Baud, 8 Daten-Bits, ungerade Parität, 1 Stopp-Bit

Beispiel-Daten: ASCII „G“ = 71 = 0100 0111 binär

Übertragung der Daten von rechts (Bit 0) nach links (Bit 7)



5 Bus-Systeme

5.3 I²C (TWI)

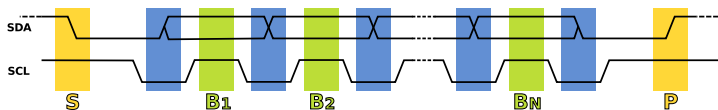
I²C = Inter-IC; TWI = Two-Wire-Interface

seriell

- *SDA*: 1 Leitung für Daten (in beiden Richtungen)
- *SCL*: Taktleitung (Clock)
- *GND*: gemeinsame Masse
- evtl. *VCC*: Stromversorgung für Peripheriegerät

synchron

- Abgleich über Taktleitung



busfähig

- *Master* initiiert Kommunikation und steuert Taktleitung
- erstes gesendetes Byte: *Adresse* des Teilnehmers
- 2 Adressen pro Teilnehmer: Lesen/Schreiben

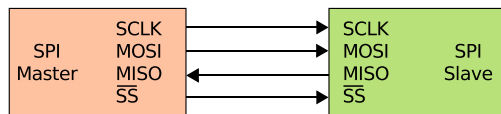
5 Bus-Systeme

5.4 SPI

Serial Peripheral Interface

seriell

- *MOSI*: Master Out, Slave In
- *MISO*: Master In, Slave Out
- *SCLK*: Taktleitung (Clock)
- \overline{SS} : Slave Select (invertiert)
- *GND*: gemeinsame Masse
- evtl. *VCC*: Stromversorgung für Peripheriegerät



synchron

- Abgleich über Taktleitung

busfähig

- *Master* initiiert Kommunikation und steuert Taktleitung
- *Slave* wird über *Slave Select* ausgewählt

5 Bus-Systeme

5.4 SPI

Serial Peripheral Interface

seriell

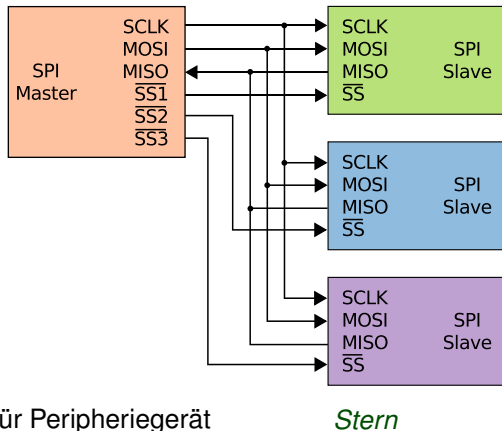
- *MOSI*: Master Out, Slave In
- *MISO*: Master In, Slave Out
- *SCLK*: Taktleitung (Clock)
- \overline{SS} : Slave Select (invertiert)
- *GND*: gemeinsame Masse
- evtl. *VCC*: Stromversorgung für Peripheriegerät

synchron

- Abgleich über Taktleitung

busfähig

- *Master* initiiert Kommunikation und steuert Taktleitung
- *Slave* wird über *Slave Select* ausgewählt



5 Bus-Systeme

5.4 SPI

Serial Peripheral Interface

seriell

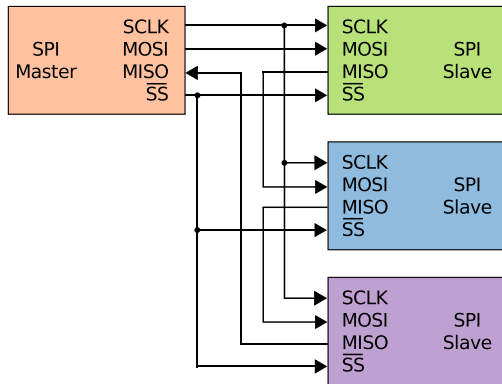
- *MOSI*: Master Out, Slave In
- *MISO*: Master In, Slave Out
- *SCLK*: Taktleitung (Clock)
- \overline{SS} : Slave Select (invertiert)
- *GND*: gemeinsame Masse
- evtl. *VCC*: Stromversorgung für Peripheriegerät

synchron

- Abgleich über Taktleitung

busfähig

- *Master* initiiert Kommunikation und steuert Taktleitung
- *Slave* wird über *Slave Select* ausgewählt



*Kaskade
Daisy Chain*

5 Bus-Systeme

5.4 SPI

Serial Peripheral Interface

seriell

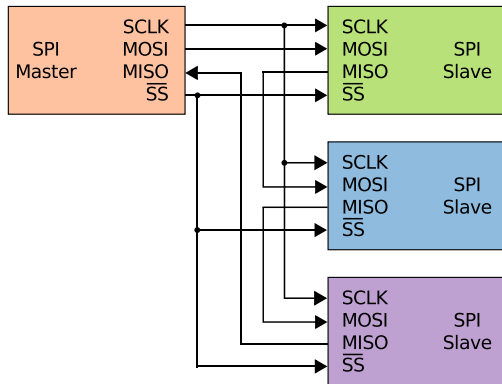
- *MOSI*: Master Out, Slave In
- *MISO*: Master In, Slave Out
- *SCLK*: Taktleitung (Clock)
- \overline{SS} : Slave Select (invertiert)
- *GND*: gemeinsame Masse
- evtl. *VCC*: Stromversorgung für Peripheriegerät

synchron

- Abgleich über Taktleitung

busfähig

- *Master* initiiert Kommunikation und steuert Taktleitung
- *Slave* wird über *Slave Select* ausgewählt



*Kaskade
Daisy Chain*

Slave gibt MOSI-Input um 1 Takt verzögert an MISO aus → Master setzt „im richtigen Moment“ \overline{SS}

5 Bus-Systeme

5.5 PWM

Pulsweitenmodulation – *pulse-width modulation*

- Steuerung von Motoren
- Nutzung als allgemeines Protokoll zur Übertragung analoger Werte

5 Bus-Systeme

5.6 Sonstiges

Matrix-Schaltung

- möglichst viele Aktoren/Sensoren
über möglichst wenige digital Inputs abfragen
bzw. über möglichst wenige digitale Outputs steuern
- Beispiele: LED-Felder, Tastaturen

R/2R-Netzwerk

- möglichst viele digitale Inputs
über einen einzigen analogen Input abfragen
- Beispiele: Tastaturen