

12 Schnittstellen, Busse & Netze

Ziele:

1. Unterscheidung zwischen Schnittstelle, Bussystem und Netz
2. Vorstellung von Verbindungsarten und Übertragungsarten
3. Anforderungen an Verbindungsleitungen
4. Vorstellung ausgewählter Schnittstellen, Bussysteme und Netze

Stichworte: Serielle und parallele Schnittstellen, Übertragungsmedien, Impedanzanpassung, Übersprechen, kapazitive und induktive Kopplung, Alien crosstalk, Abschirmung, symmetrische und asymmetrische Übertragung, Twisted-Pair, Software- und Hardware-Handshake, RS-232, V.24, IEEE 1284, SPI, GPIB, USB und Ethernet

Lehrbücher:

- P. Horowitz und W. Hill [2]
- E. Hering, K. Bresser, J. Gutekunst [3]

Warnung: Keine Gewähr auf spezifische Angaben in diesem Kapitel. Bitte überprüfen Sie Daten zu Bauelementen, Steckern, Pinbelegungen etc. bevor Sie diese anwenden.

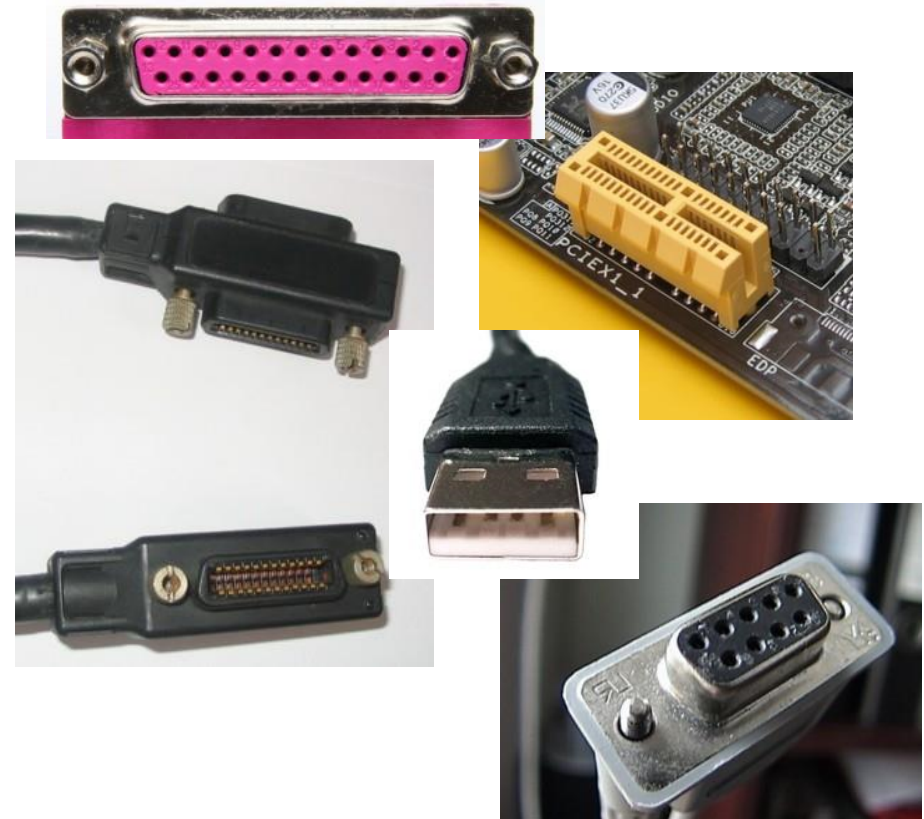


Abb. 12.1: Der Schnittstellen- und Stecker-Zoo.

12.1 Grundbegriffe der Datenübertragung

Schnittstelle:

- Verbindung zweier Systeme (Punkt zu Punkt)
- über Schnittstellenleitung
- charakteristisch für jede Schnittstelle:
 - Signale (Dauer, Pegel) und ihre Bedeutung
 - Zahl der Leitungen und Bezeichnung
 - elektrische, mechanische und funktionale Eigenschaften

Bussystem:

- Verbindungssystem zwischen mehreren Schaltwerken
- gemeinsame Nutzung des Übertragungswegs
- Adressierung der Teilnehmer
- parallele Bussysteme
- serielle Bussysteme
 - einfacher Aufbau der Busstruktur
 - kostengünstige Implementierung
 - gewinnen an Bedeutung (USB)

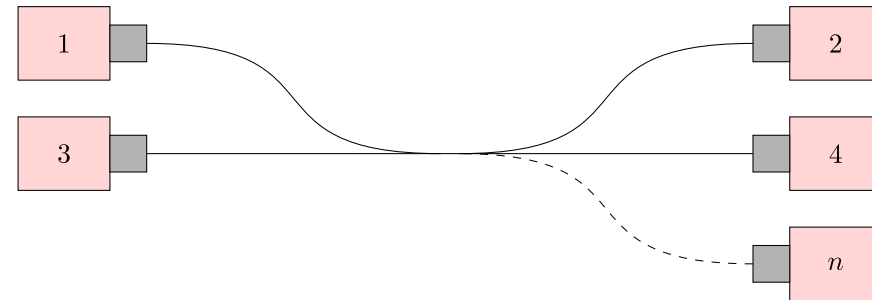
Netz:

- inhomogene Architektur über Busgrenzen hinweg
- meist lose gekoppelte Systeme
- räumlich getrennte Rechner
- unterschiedliche Schnittstellen
- unterschiedliche Protokolle im gleichen Netz

a) Point-to point interface



b) Bus architecture



c) Network

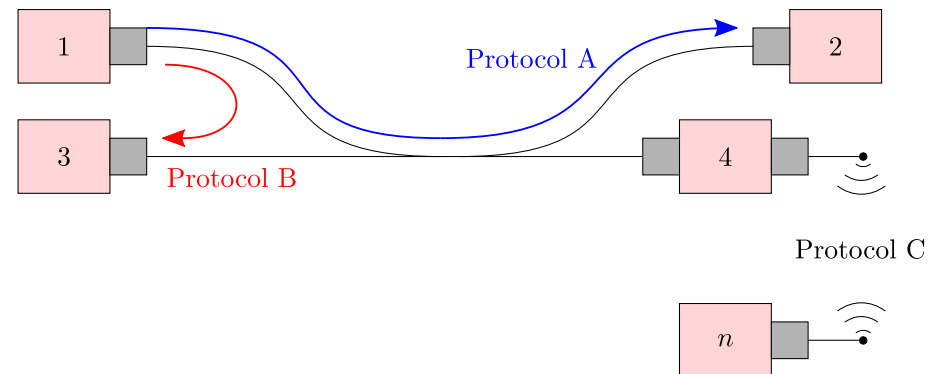


Abb. 12.2: a) Punkt-zu-Punkt-Schnittstelle, b) Bussystem und c) Netz.

12.1.1 Verbindungsarten

Punkt-zu-Punkt: (peer to peer)

- Rechner und Drucker
- Rechner und Messgerät
- häufig RS-232

Punkt zu Gruppe: (multicast)

- Leitrechner zu Steuergeräten
- Rechner zu VPNs
- PC über USB zu Laborgeräten

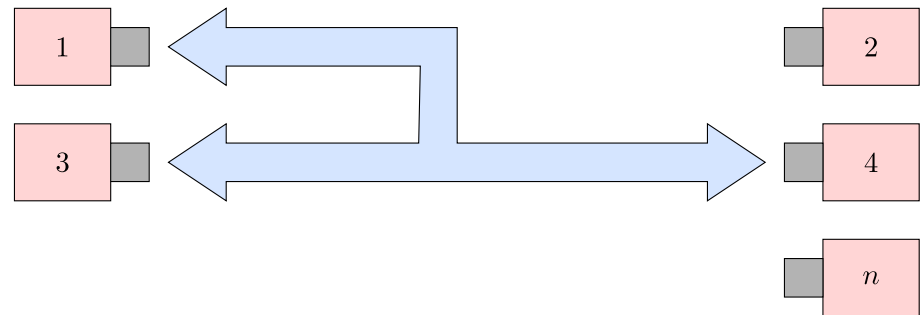
Punkt zu allen: (broadcast)

- Systemmanager an alle Teilnehmer
- Telekommunikation

a) Peer to peer



b) Multicast



c) Broadcast

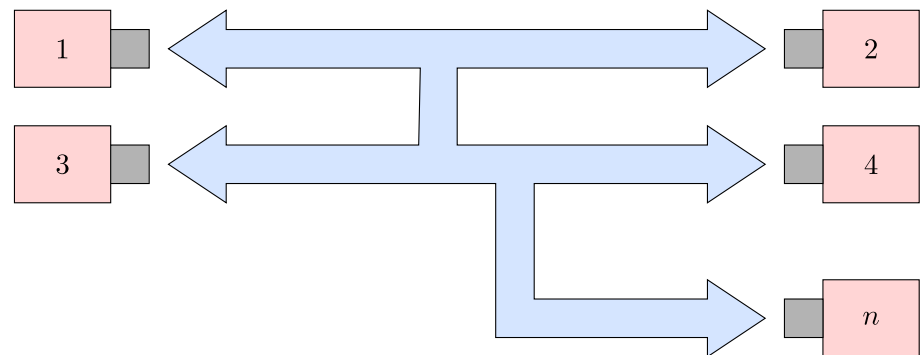


Abb. 12.3: Verbindungsarten. Nach Ref. [3], S. 684.

12.1.2 Übertragungsarten

Es gibt grundsätzlich zwei.

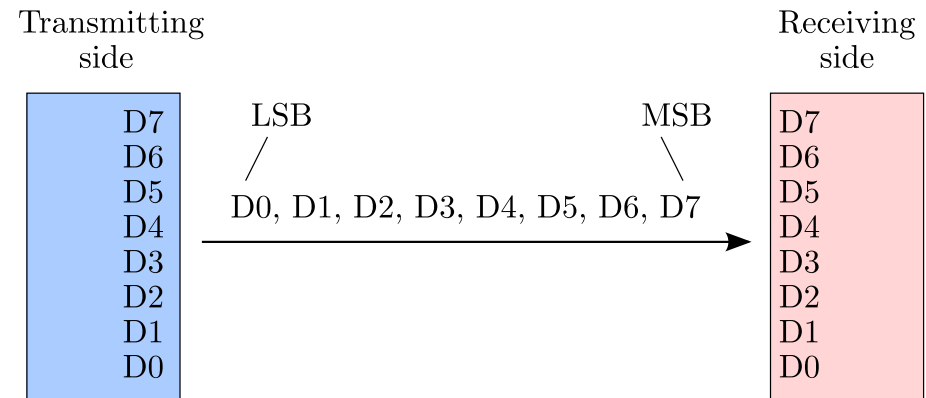
Serielle Übertragung:

- bitweise Übertragung
- Beispiele:
 - RS-232
 - USB
 - Ethernet
 - Fire-Wire
 - Serial ATA

Parallele Übertragung:

- gleichzeitige Übertragung mehrerer Bits
- meist ganze Wörter, Bytes
- Beispiele:
 - IEEE 1284 (alias printer port)
 - IEC-625-Bus (alias IEEE-488, GPIB)
 - Bussystem im PC (DRAM)

a) Serial interface



b) Parallel interface

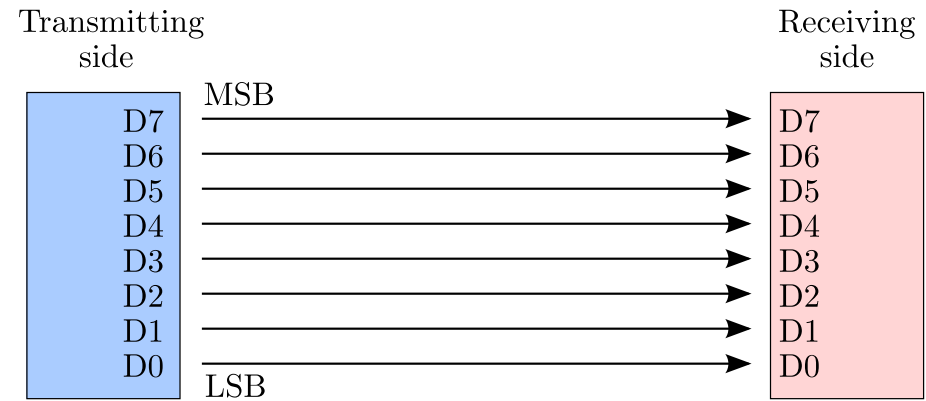


Abb. 12.4: a) Serielle Übertragung, b) parallele Übertragung.

12.1.3 Ansprüche an Datenübertragung:

- störfrei
- schnell
- große Distanzen
- geringe Leistungsaufnahme
- möglichst einfach & kostengünstig
- leicht handzuhaben

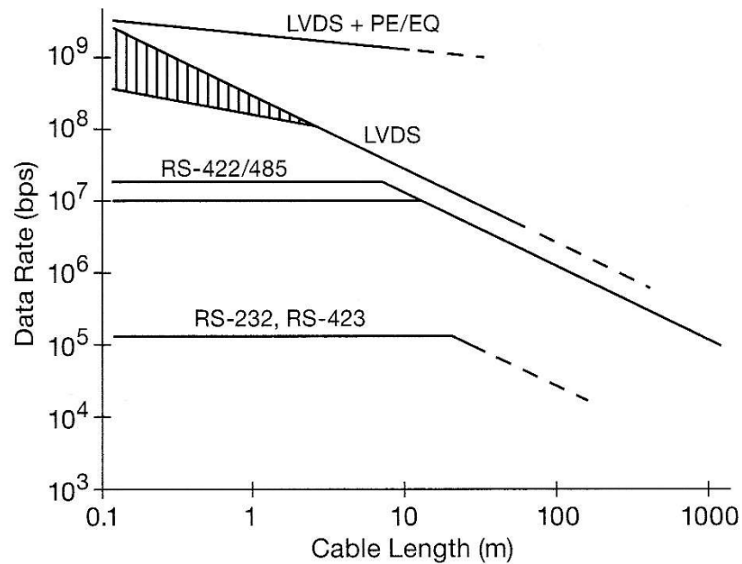


Abb. 12.5: Übertragungsraten in Abhängigkeit von der Kabellänge. Quelle: Horowitz & Hill, S. 873 [2].

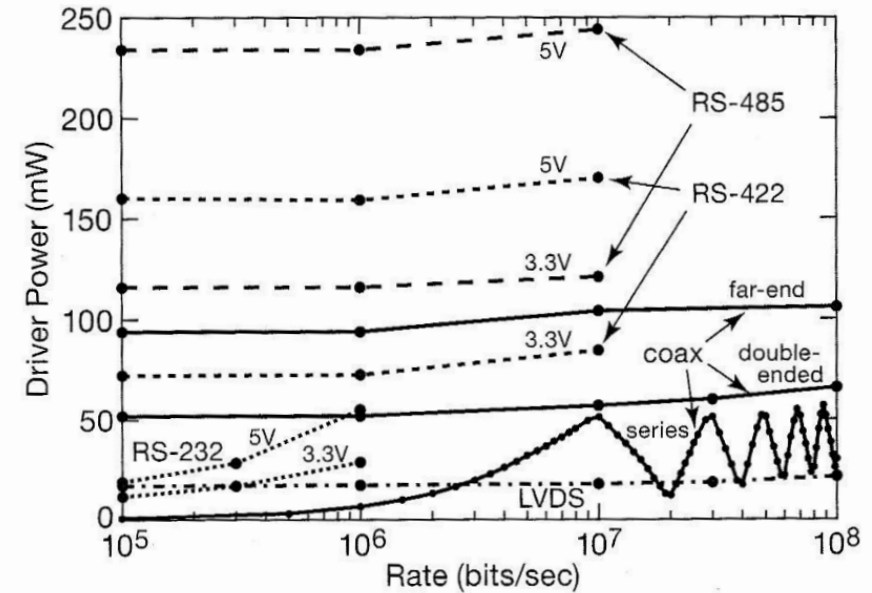


Abb. 12.6: Leistungsaufnahme in Abhängigkeit von der Datenrate. Quelle: Horowitz & Hill, S. 874 [2].

12.2 Verbindungsleitungen

12.2.1 Übertragungsmedien

leitergebunden	elektrisch	koaxial	$< 10^5$ Hz
		verdrillt	$< 10^5$ Hz
		Wellenleiter	abh. von Abmess.
Lichtwellenleiter		Monomode	$< 10^{16}$ Hz
		Multimode	$< 10^{16}$ Hz
frei	Funk		$10^5 - 10^{16}$ Hz
	Infrarot		$10^{12} - 10^{15}$ Hz
	Mikrowelle		$10^{10} - 10^{12}$ Hz

Tab. 12.1: Eigenschaften verschiedener Übertragungsmedien

Quelle: G. Dreo Rodosek, Leibniz-Rechenzentrum.

Übertragung mittels Trägersignal:

- Funk, Infrarot & Mikrowelle
- aber auch beim elektrischen Wellenleiter
- Bedingung:

$$\omega_s \ll \omega_c \quad (10)$$

Im Folgenden:

- Fokus auf elektrische Übertragungen
- interessante Größen: $V(t)$, $I(t)$

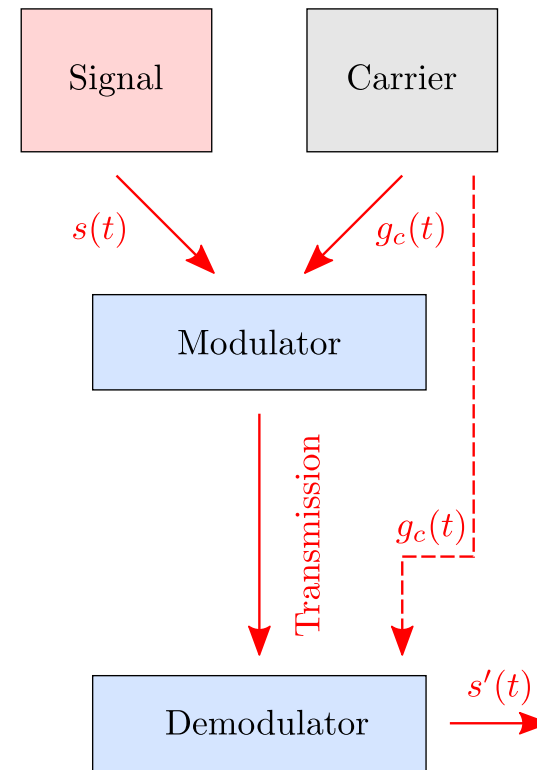


Abb. 12.7: Übertragung mittels Trägersignal.

12.2.2 Störung der Übertragung durch Reflexionen

Reflexionen: z.B bei einem Koaxial-Kabel

- BNC (Bayonet Neill Concelman)
- meist Wellenwiderstand $Z = 50 \Omega$
- im Labor: analoge Signalübertragung

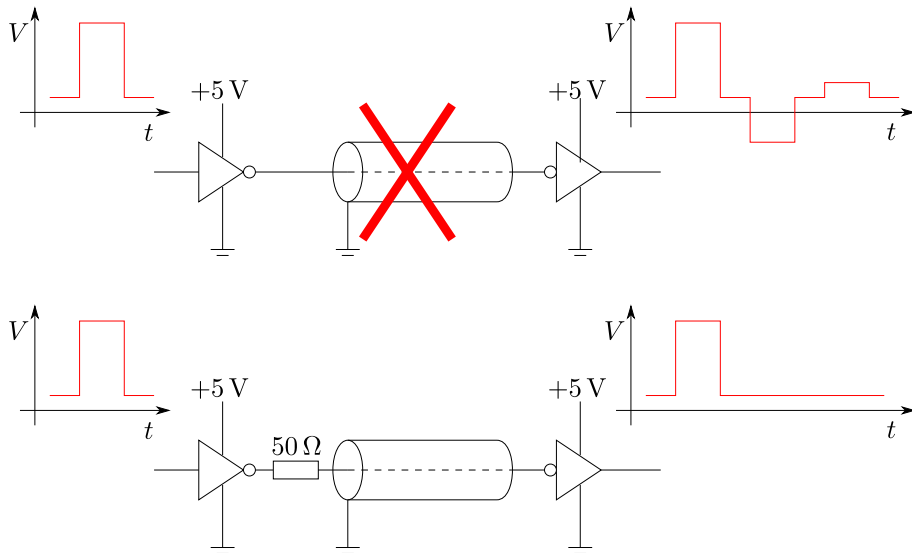


Abb. 12.8: Vermeidung von Reflexionen, wenn das Kabel entsprechend terminiert wird.

Reflexionsfaktor und Reflexion:

$$r = \frac{Z_1 - Z_2}{Z_1 + Z_2} \quad \text{und} \quad R = \frac{(Z_1 - Z_2)^2}{(Z_1 + Z_2)^2} \quad (11)$$



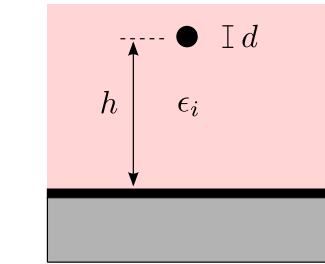
Abb. 12.9: BNC-Stecker eines Koaxialkabels. Quelle: Wikipedia.

Standard	$Z(\Omega)$	Tolerance
Ethernet Cat.5	100	$\pm 5 \Omega$
USB	90	$\pm 15\%$
HDMI	95	$\pm 15\%$
IEEE 1394	108	+3% -2%
VGA	75	$\pm 5\%$
DisplayPort	100	$\pm 20\%$
DVI	95	$\pm 15\%$
PCIe	85	$\pm 15\%$

Tab. 12.2: Wellenwiderstände verschiedener Standards. Quelle: Wikipedia

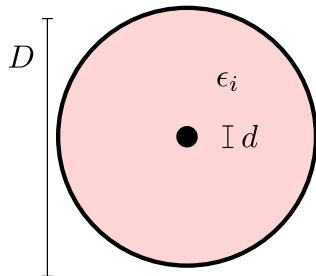
Berechnung von Wellenwiderständen:

www.eeweb.com/tools/symmetric-stripline-impedance



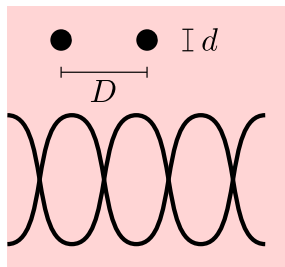
a) Wire microstrip

$$Z = \frac{60 \Omega}{\sqrt{\epsilon_i}} \ln \left(\frac{4h}{d} \right)$$



b) Coaxial cable

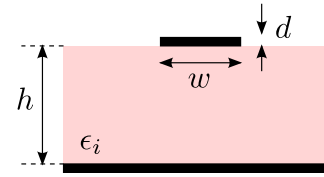
$$Z = \frac{60 \Omega}{\sqrt{\epsilon_i}} \ln \left(\frac{D}{d} \right)$$



c) Twisted pair

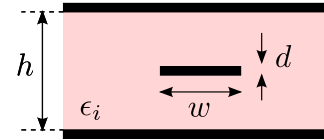
$$Z = \frac{120 \Omega}{\sqrt{\epsilon_i}} \ln \left(\frac{2D}{d} \right)$$

Abb. 12.10: Wellenwiderstände von Leitungen.



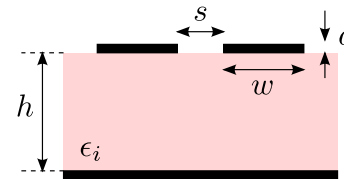
a) Microstrip

$$Z = \frac{60 \Omega}{\sqrt{0.457 \epsilon_i + 0.67}} \ln \left(\frac{4h}{0.067(0.8w+d)} \right)$$



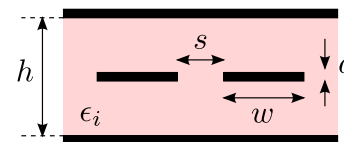
b) Stripline

$$Z = \frac{60 \Omega}{\sqrt{\epsilon_i}} \ln \left(\frac{4h}{0.067 \pi(0.8w+d)} \right)$$



c) Microstrips
differential mode

$$Z_{diff} = 2 Z (1 - 0.48 \exp \{-0.96 s/h\})$$



d) Striplines
differential mode

$$Z_{diff} = 2 Z (1 - 0.374 \exp \{-2.9 s/h\})$$

Abb. 12.11: Wellenwiderstände von Microstrips und Striplines. Nach Ref. [1].

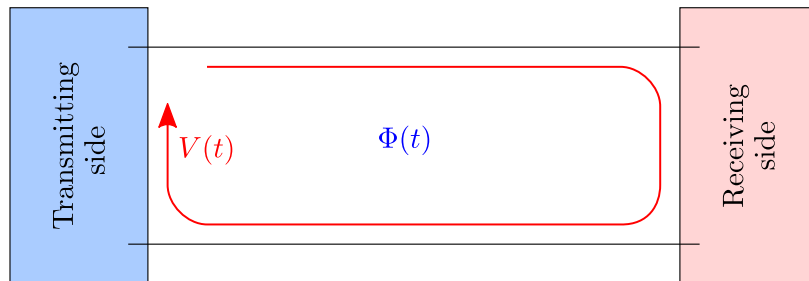
Striplines & Coax: Dispersionsfrei, wenn $\epsilon_i = const$

12.2.3 Störungen durch elektromagn. Einstrahlung

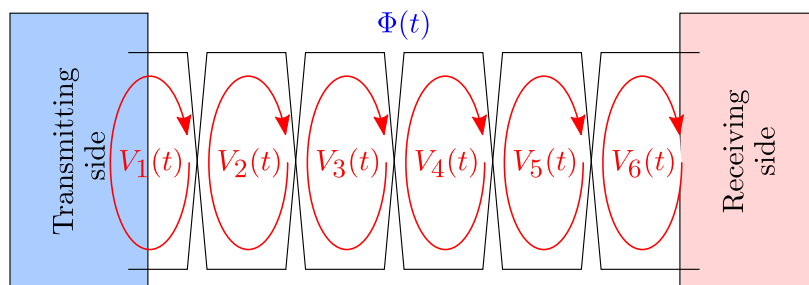
- engl.: electromagnetic interference
- Ursache: magnetische Induktion $\vec{B}(t)$
- Faradaysches Induktionsgesetz:

$$V(t) = \oint \vec{E} \cdot d\vec{s} = -\frac{d}{dt}\Phi_m(t) \quad (12)$$

a) Two-wire line



b) Twisted pair cable



$$V(t) = \sum_i V_i(t) \approx 0$$

Reduzierung des Rauschens:

- Verdrillung zum Twisted Pair
- zusätzliche äußere Abschirmung (shielded twisted Pair, STP)
 - Ethernet: Cat7 –Cat8
- ohne weitere Abschirmung (unshielded twisted Pair, UTP)
 - Ethernet: Cat3 –Cat6

Quelle	Frequenz [MHz]
Radio-Rundfunk	0.15 – 110
Fernsehfunk	60 – 800
D-Netz	790 – 960
E-Plus	1710 – 1880
UMTS	1920 - 2170
Schnurlose Telefone	1880 – 1900
Bluetooth	2400 – 2450 MHz
Wireless LAN	2400 oder 5800
Mikrowellenherde	um 3000

Tab. 12.3: Einige Quellen elektromagnetischer Strahlung und zugehörige Frequenzen.

Abb. 12.12: a) Zweiadriges Kabel. b) Verdrilltes Kabel.

12.2.4 Störungen durch Übersprechen

- engl.: crosstalk
- von einem Leiter auf den anderen

Kapazitive Kopplung:

- benachbarte Adern wirken wie parasitäre Kapazitäten
- besonders stark bei:
 - hohen Pegeln
 - großen Leitungsimpedanzen

Induktive Kopplung:

- benachbarte Adern wirken wie benachbarte Spulen
- besonders stark bei:
 - großen Strömen
 - hohen Frequenzen
 - kleinen Leitungsimpedanzen

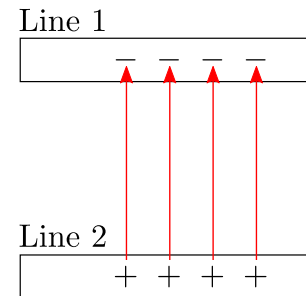
Im Labor: Alien crosstalk

- zwischen Kabeln zu verschiedenen Instrumenten

Strategie zur Vermeidung:

- Abschirmung mit Metallfolie oder Drahtgeflecht
- wirkt wie ein Faraday-Käfig
- reduziert kapazitives Koppeln
- reduziert induktives Koppeln bei großen Frequenzen

a) Capacitive crosstalk



b) Shielded

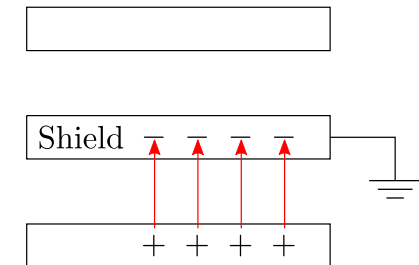


Abb. 12.13: Abschirmung des kapazitiven Übersprechens.

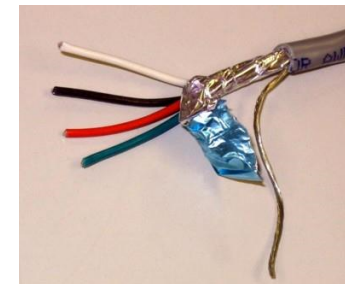


Abb. 12.14: Durch metallisierte Folie abgeschirmtes Kabel.

Skin-depth:

- Maß für die Eindringtiefe elektromagn. Felder

$$\delta(\nu) \approx \sqrt{\frac{\rho}{\pi \nu \mu \mu_0}} \quad (13)$$

- mit spezifischem Widerstand ρ , magnetischer Permeabilität μ und magnetischer Feldkonstante μ_0

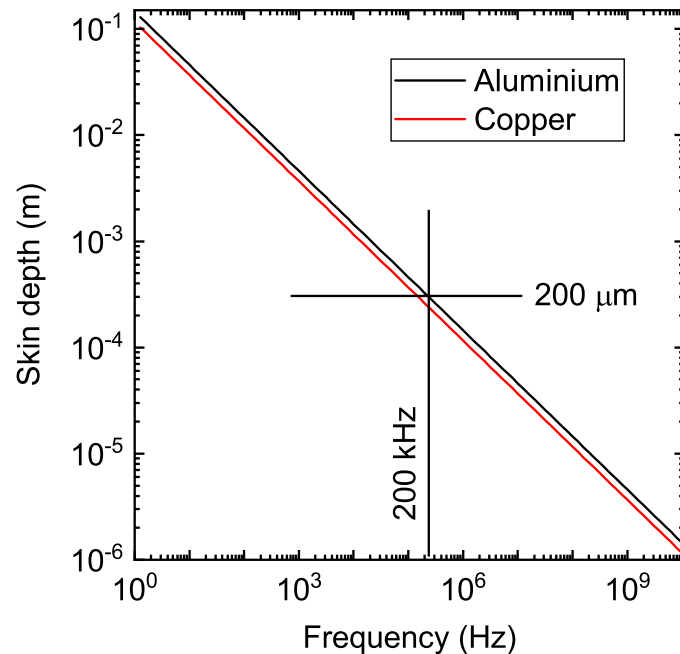


Abb. 12.15: Frequenzabhängigkeit der Abschirmtiefe für Aluminium und Kupfer.

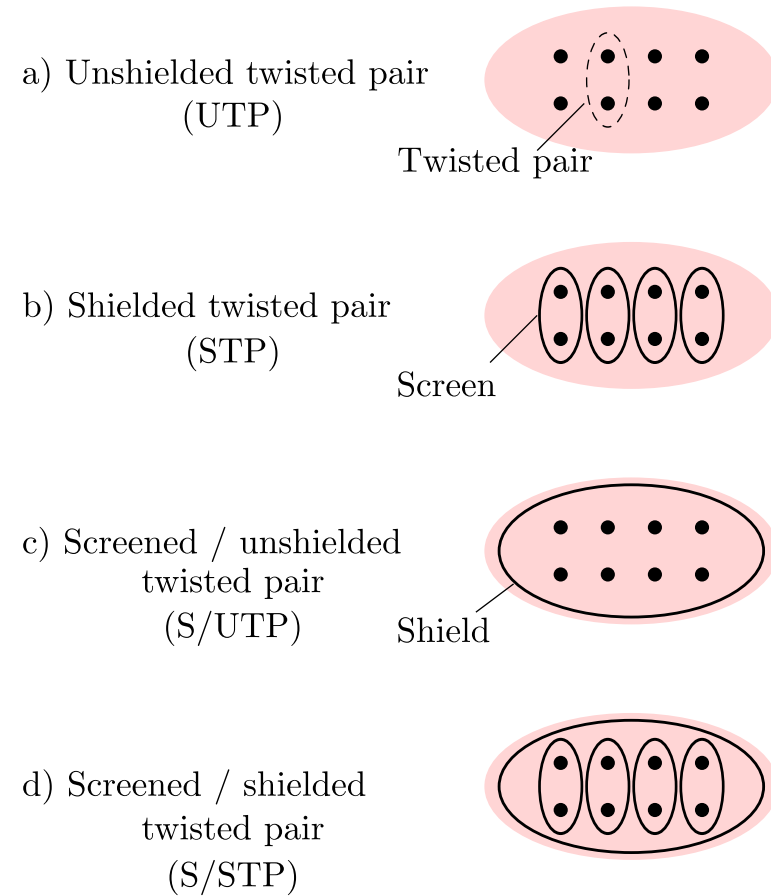


Abb. 12.16: Ausführungen von verdrehten Kabeln.

Asymmetrische Signalübertragung:

- Signalspannung gegenüber gemeinsamer Masse
- Vorteil: nur eine Masse für alle Signalleitungen
- Nachteil: anfällig gegenüber Störungen von außen
- mögliche Verbesserungen:
 - Masseleitung zwischen jeder Signalleitung (reduziert Übersprechen)
 - Abschirmung (reduziert Störungen von außen)

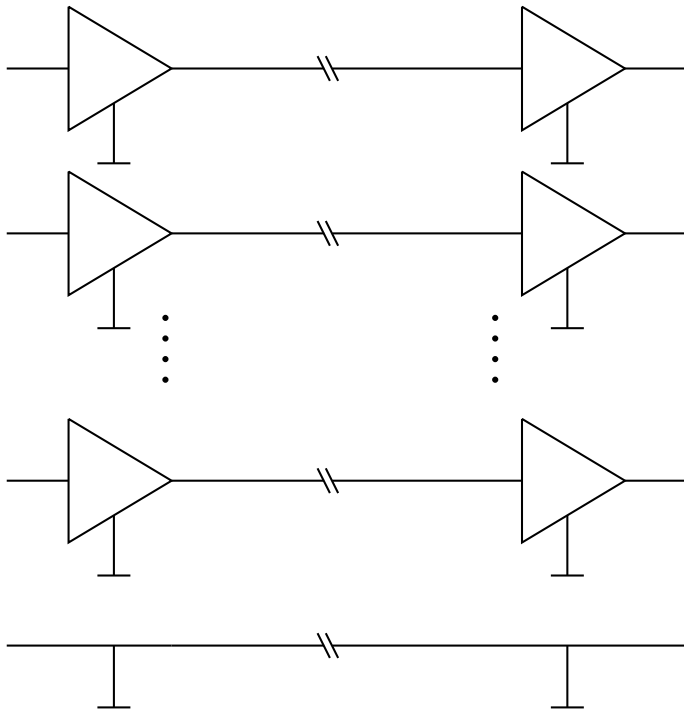


Abb. 12.17: Schema zur asymmetrischen Übertragung.

Symmetrische Signalübertragung:

- auch differentielle Übertragung
- jedes Signal auf zwei Leitungen
- mit gegensätzlicher Polarität
- beide Leitungen nehmen das Störsignal auf
- Differenz eliminiert Störsignal
- erhöht den Störabstand
- Bandbreite: größer als 100 Mbits/s
- Nachteil: große Anzahl von Leitungen

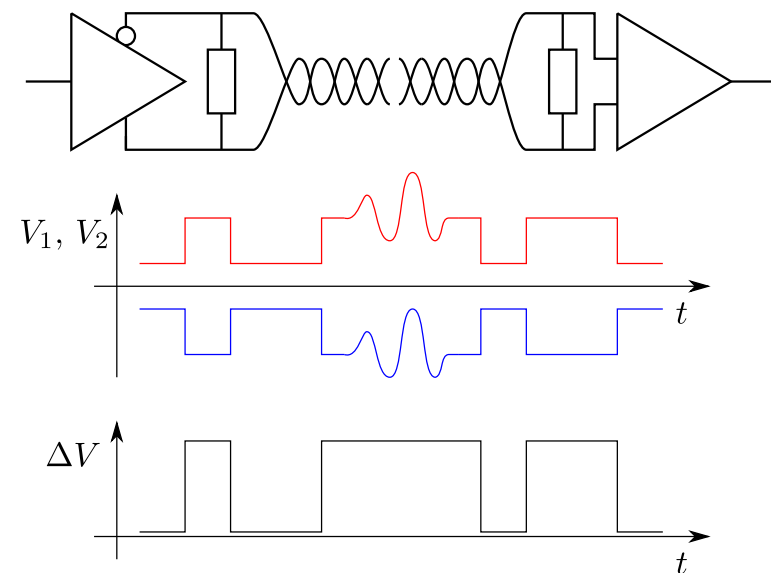


Abb. 12.18: Schema zur symmetrischen (differentiellen) Übertragung.

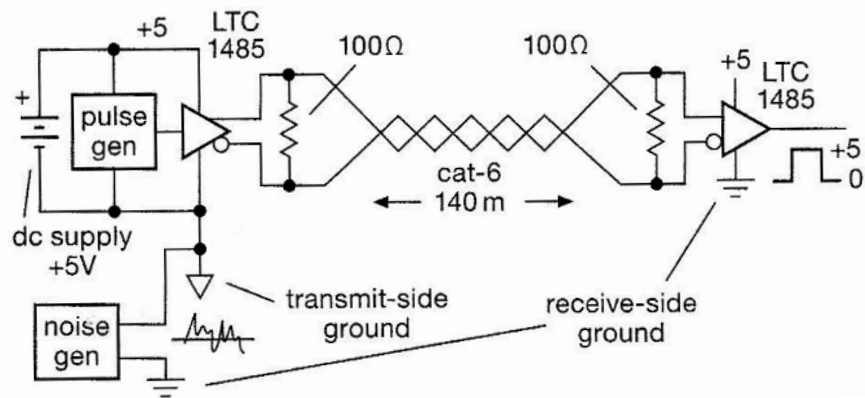


Abb. 12.19: Aufbau eines „Torture tests“ zur differentiellen Übertragung. Quelle: Horowitz & Hill [2], S. 120. Der Treiber ist „float“ und wird mit dem „Noise generator“ mit 15 Vpp gegenüber der Masse des Empfängers belastet.

LTC1485:

- low power differential bus/line transceiver
- driver propagation delay: 28 ns
- Kosten 2.80 – 3.50 EUR
- als 8-Pin DIP
- MultiSim: leider keine entsprechenden Bauteile

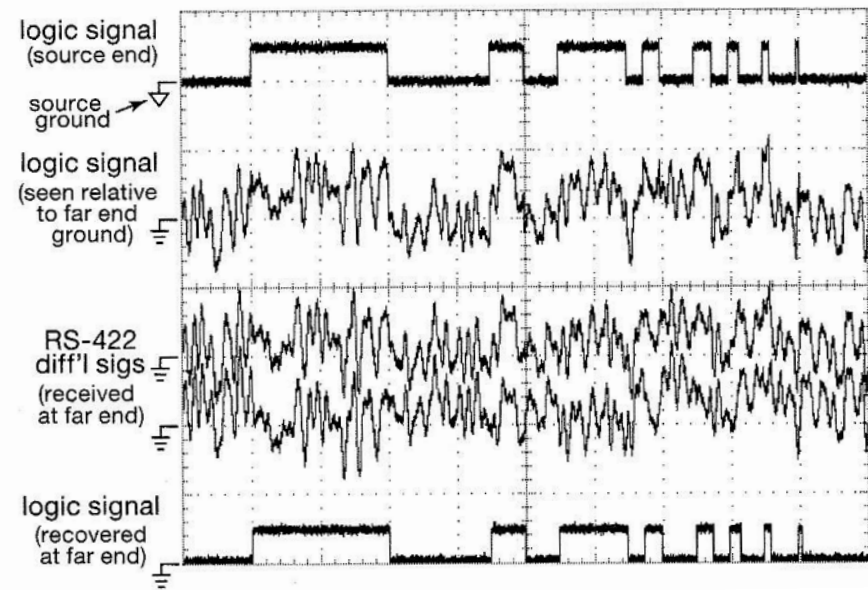


Abb. 12.20: Ergebnis des „Torture tests“ zur differentiellen Übertragung. Quelle: Horowitz & Hill [2], S. 120. Die horizontale Skalierung beträgt 10 V/div und die vertikale beträgt 100 μ s/div.

12.3 Synchronisation

Zu vermitteln ist:

- Datenübertragungsrate
- sowie Beginn und Ende eines Datenpakets

Asynchrone Übertragung:

- Datenstrom kann jederzeit einsetzen
- Anfang und Ende sind zu markieren

Synchrone Übertragung:

- Übertragung nur zu festen Zeitpunkten
- Synchronisation auch wenn keine Daten vorliegen

	asynchrone Übertragung	synchrone Übertragung
parallel	Handshake-betrieb	gemeinsame Taktleitung
seriell	Start-Stop-Verfahren	Leitungscode oder Taktleitung

Tab. 12.4: Übertragungsarten und technische Realisierungen.

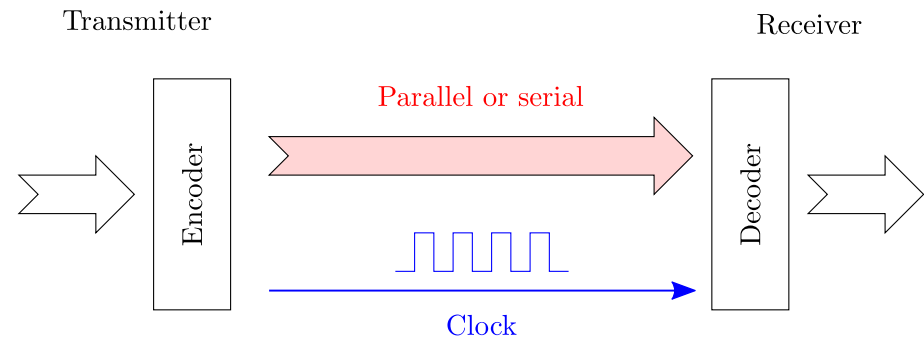


Abb. 12.21: Schema zu einer synchronen Übertragung (seriell sowie parallel), wenn eine gemeinsame Taktleitung vorliegt.

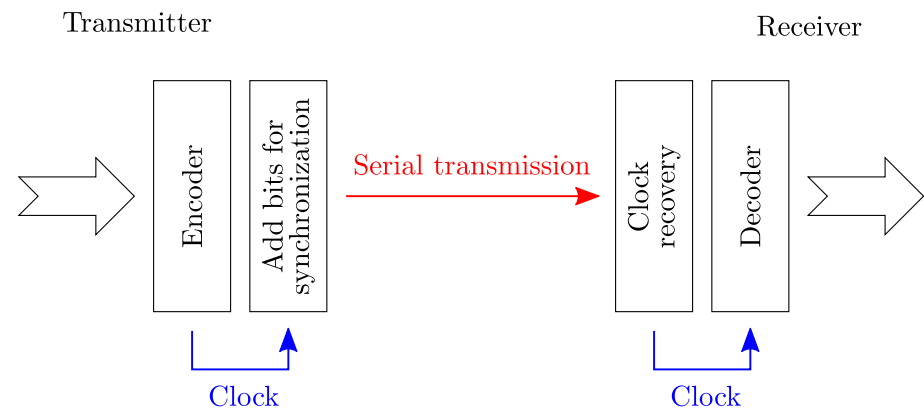


Abb. 12.22: Schema zu einer seriellen asynchronen Übertragung.

Software-Handshake:

- insgesamt nur eine Datenleitung für:
 - Daten
 - Steuersignale, bidirektional
- Kommunikation: zeitlich sequentiell

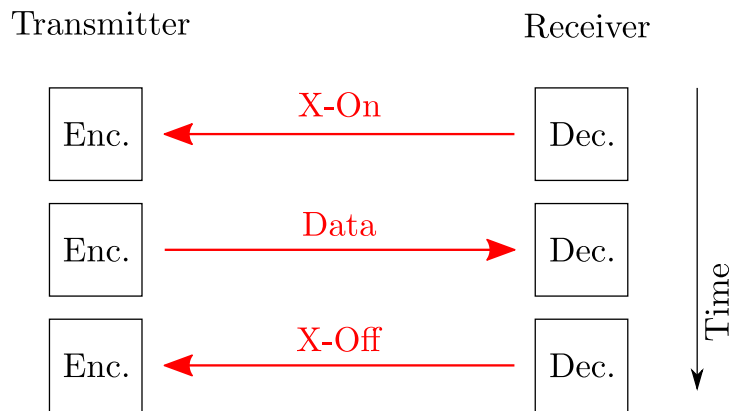


Abb. 12.23: Schnittstellen-unspezifisches Schema zu einem simplen Software-Handshake. Die Befehle X-On und X-Off stehen für „transmitter on/off“.

Hardware-Handshake:

- eine Datenleitung
- mindestens zwei Steuerleitungen
- Befehle:
 - request to send (RTS)
 - clear to send (TS)

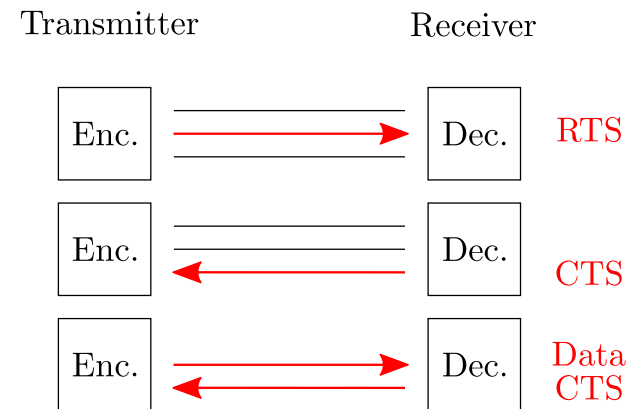


Abb. 12.24: Schnittstellen-unspezifisches Schema zu einem möglichen Hardware-Handshake. Die Befehle RTS und CTS stehen für „request to send“ und für „clear to send“.

12.4 RS-232 oder V.24

Standards:

- amerikanische Norm: RS-232C
- in Europa: fast indentisch V.24

Kommunikationsgeräte:

- Datenquelle, „transmitter“, TX
- Datensenke, „receiver“, RX

Kenngößen:

- Signalpegel:
 - Low: +3 V – +15 V
 - High: –3 V – –15 V
 - gutes Signal/Störverhältnis
 - 15 m Übertragungsdistanz nach Norm
- Steuerpegel: umgekehrte Polarität von Low & High
- Maximal 115.200 Baud bei Längen unter 2 m

Wichtigste Leitungen: weitere für Telekommunikation

- GND: Massepotential für Datenleitungen
- TXD: Sendedaten vom Computer zum Peripheriegerät
- RXD: Sendedaten vom Peripheriegerät zum Computer
- RTS: Übertragungsbereitschaft des Computers
- CTS: Empfangsbereitschaft des Peripheriegeräts

Bedeutung: Labortechnik

DIN	US	25-pol.	9-pol.	Beschreibung	Richtung PC ... Periph.
E1	–	1	–	Protective ground	–
E2	GND	7	5	Common ground	–
D1	TXD	2	3	Transmitted data	→
D2	RXD	3	2	Received data	←
S2	RTS	4	7	Request to send	→
M2	CTS	5	8	Clear to send	←
M1	DSR	6	6	Data set ready	←
S1.2	DTR	20	4	Data terminal ready	→
M5	DCD	8	1	Data carrier detect	←
M3	RI	22	9	Ring indicator	←

Tab. 12.5: Pin-Belegungen bei der Schnittstelle RS-232.

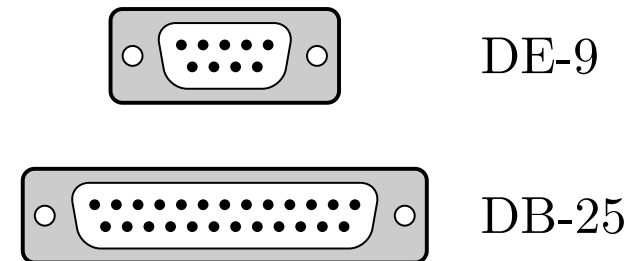
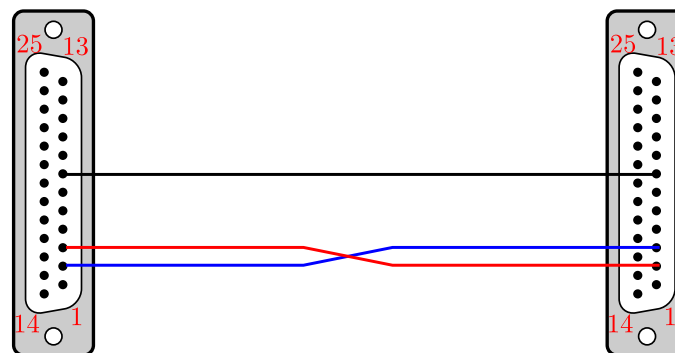
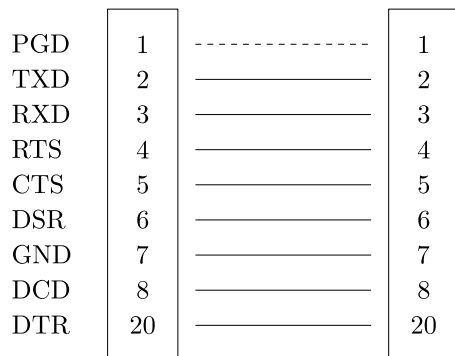


Abb. 12.25: Die beliebtesten D-Sub-Steckverbindungen.

a) Computer to peripheral device



b) Null modem

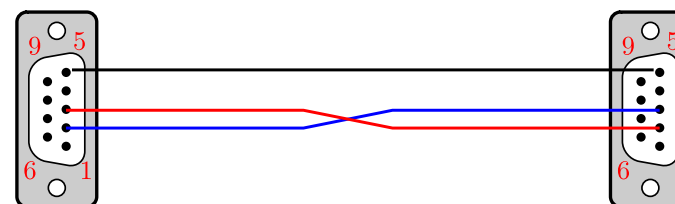
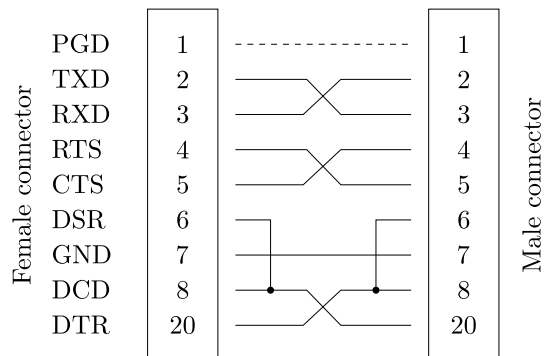


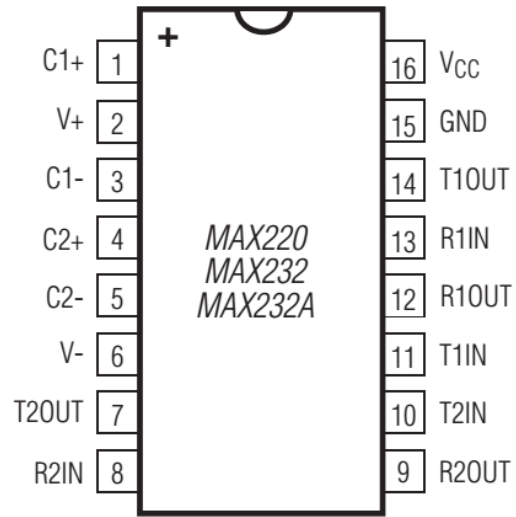
Abb. 12.26: Belegung von DB-25-Verbindern. a) für eine Verbindung zwischen Computer und Peripheriegerät. b) ein mögliches Null-Modem für die Verbindung zwischen zwei Computern.

Abb. 12.27: Null-Modem ohne Hardware Handshaking.

Null-Modem:

- Verbindung zwischen zwei Computern
- Verbindung zur selben Schnittstelle zu Testzwecken
- Vielzahl von verschiedenen Null-Modems
- einfachste Form: s. Abb. 12.27
 - ohne Hardware-Handshake
 - drei Kabel

TOP VIEW



DIP/SO

CAPACITANCE (μF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

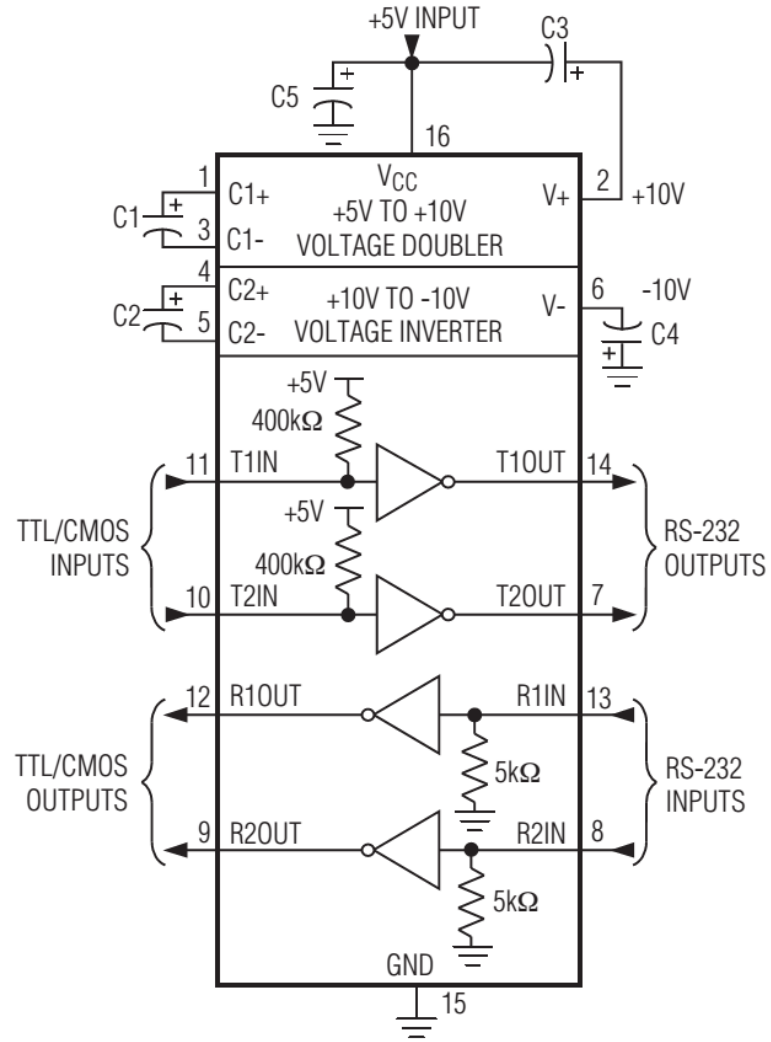


Abb. 12.28: Die Maxim220/232-Serie, mit der RS-232-Schnittstellen realisiert werden können.

12.5 Parallele Schnittstellen & IEEE 1284

Verbindung:

- vom Computer zu einem Peripheriegerät
- oder mehreren (Einschleifen des Kabels)

Ursprünglich:

- unidirektionale Kommunikation
- Centronics-Schnittstelle
- Drucker-Schnittstelle
- line printing port (LPT, LPT1, LPT2, ...)

Standard seit 1994:

- IEEE 1284
- bidirektionale Kommunikation
- bis zu 2 Mbit/s pro Richtung
- jedes Byte wird durch Handshake synchronisiert
- benötigt Schnittstellenkarte / Interface

Geräte mit paralleler Schnittstelle:

- Iomega Zip 100
- HP C4381A CD-Writer ...
- heute keine Bedeutung bei Labortechnik
- ersetzt durch USB

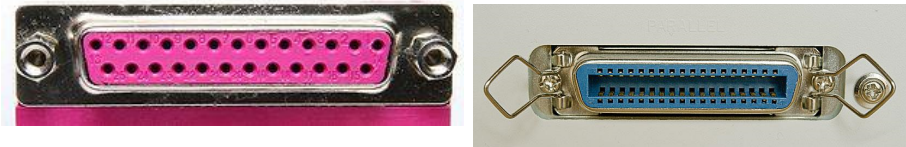


Abb. 12.29: Links: Parallele Schnittstelle auf der Hauptplatine eines PCs. Rechts: Centronics-Schnittstelle eines Peripheriegeräts. Quelle: Wikipedia.

Pin No. (DB25)	Pin No. (36 pin)	Signal name	Direction	Register bit	Inverted
1	1	Strobe	In/Out	Control-0	Yes
2	2	Data0	Out	Data-0	No
3	3	Data1	Out	Data-1	No
4	4	Data2	Out	Data-2	No
5	5	Data3	Out	Data-3	No
6	6	Data4	Out	Data-4	No
7	7	Data5	Out	Data-5	No
8	8	Data6	Out	Data-6	No
9	9	Data7	Out	Data-7	No
10	10	Ack	In	Status-6	No
11	11	Busy	In	Status-7	Yes
12	12	Paper-Out	In	Status-5	No
13	13	Select	In	Status-4	No
14	14	Linefeed	In/Out	Control-1	Yes
15	32	Error	In	Status-3	No
16	31	Reset	In/Out	Control-2	No
17	36	Select-Printer	In/Out	Control-3	Yes
18-25	19-30,33,17,16	Ground	-	-	-

Tab. 12.6: Pin-Belegung nach IEEE 1284.

12.6 Serial peripheral interface (SPI)

Eigenschaften:

- Chip-Schnittstelle
- Verbindungen innerhalb eines Geräts
- voll duplex (gleichzeitig bidirektional)
- ein Master
- eine Slave oder mehrere Slaves

Leitungen:

- MISO: Master in – slave out (oder SDO)
- MOSI: Master out – slave in (oder SDI)
- SCLK: Serial clock (oder CLK)

Parameter:

- CPOL: Taktpolarität
- CPHA: Taktphase

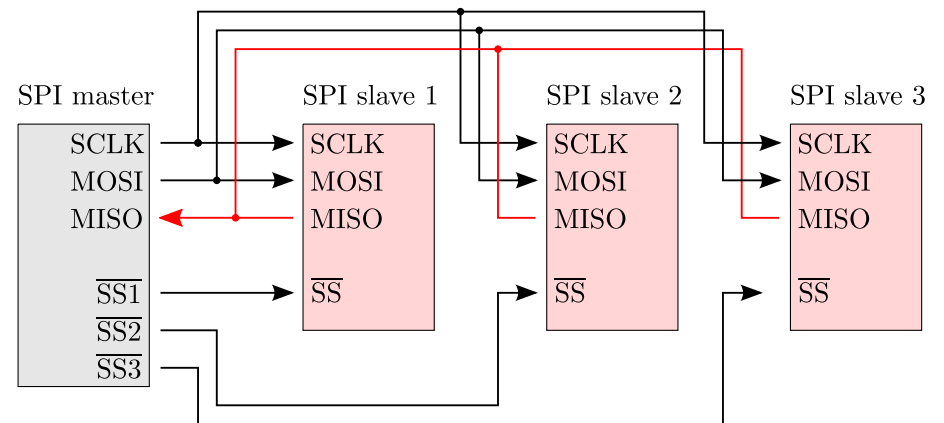
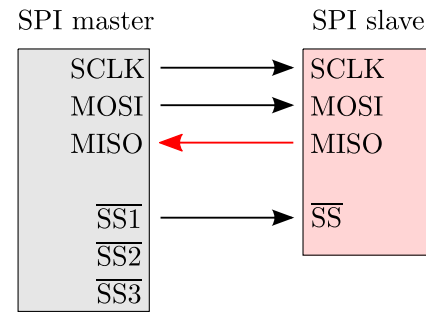


Abb. 12.30: Oben: SPI-Verbindung zwischen Master und einem Slave. Unten: Sternförmige SPI-Verbindungen um den Master herum. Nach Ref. [3], S. 703.

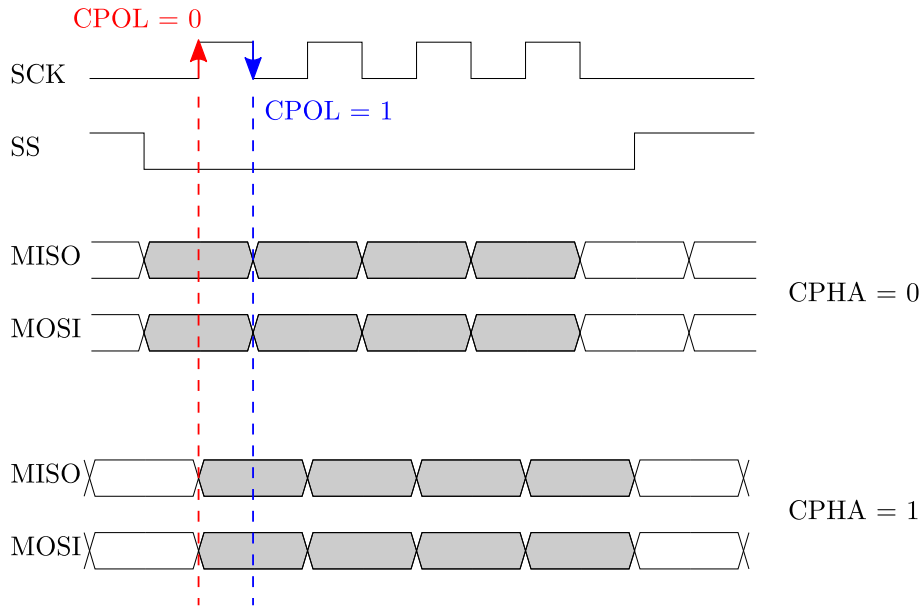


Abb. 12.31: Zur Synchronisation von SPI-Schnittstellen.
Nach Ref. [3], S. 704.

SPI-Bausteine:

- Analog-Digital- und Digital-Analog-Wandler
- EEPROM- und FLASH-Speicher (s. Abschnitt 9.5.2)
- Real Time Clocks (RTC)
- Signalmixer
- LCD-Controller
- Universal Asynchronous Receiver Transmitter (UART)
- Sensoren
- ...

- 256 taps for each potentiometer
- Potentiometer values for 10 kΩ, 50 kΩ and 100 kΩ
- Single and dual versions
- SPI™ serial interface (mode 0,0 and 1,1)
- ±1 LSB max INL & DNL
- Low power CMOS technology
- 1 μA maximum supply current in static operation

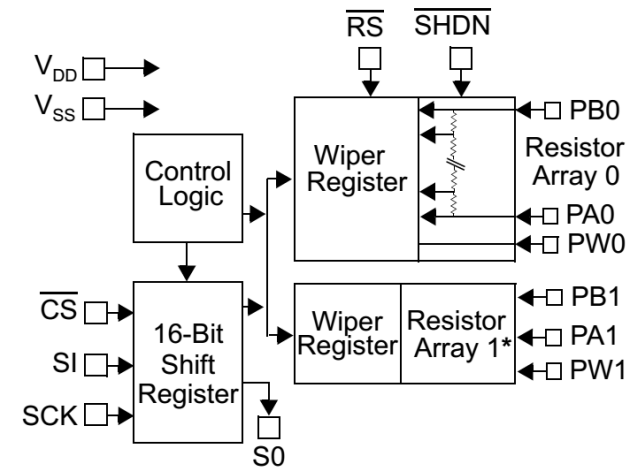
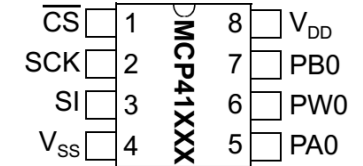


Abb. 12.32: Auszug aus dem Datenblatt zum Digitalpotentiometer MCP41100-I/P von Microchip.

12.7 IEEE 488 oder GPIB

- Wurzeln in 1960'ern
- Hewlett Packard
- Ziel: Verbindung von/zu Messgeräten
- später IEEE 488-Norm

Eigenschaften:

- Bus: 8 bit parallel
- jedes Gerät hat eine Adressnummer zwischen 0 und 30
- Topologie des Netzes:
 - linear
 - verzweigt
 - oder sternförmig
 - aber Gesamtlänge < 20 m
- Stecker können ineinander gesteckt werden

Anwendung:

- Labortechnik
- zuverlässig (Stecker sichern)
- aber hohe Kosten
 - eine Schnittstelle mit Logik ca. 500 EUR
 - Kabel, 8 m ca. 180 EUR



Abb. 12.33: GPIB-Stecker. Quelle: Wikipedia.

Pin	Short	Description
1	DIO1	Data input/output bit
2	DIO2	Data input/output bit
3	DIO3	Data input/output bit
4	DIO4	Data input/output bit
5	EOI	End-or-identify
6	DAV	Data valid
7	NRFD	Not ready for data
8	NDAC	Not data accepted
9	IFC	Interface clear
10	SRQ	Service request
11	ATN	Attention
12	SHIELD	Shield
13	DIO5	Data input/output bit
14	DIO6	Data input/output bit
15	DIO7	Data input/output bit
16	DIO8	Data input/output bit
17	REN	Remote enable
18	GND	(wire twisted with DAV)
19	GND	(wire twisted with NRFD)
20	GND	(wire twisted with NDAC)
21	GND	(wire twisted with IFC)
22	GND	(wire twisted with SRQ)
23	GND	(wire twisted with ATN)
24		Logic ground

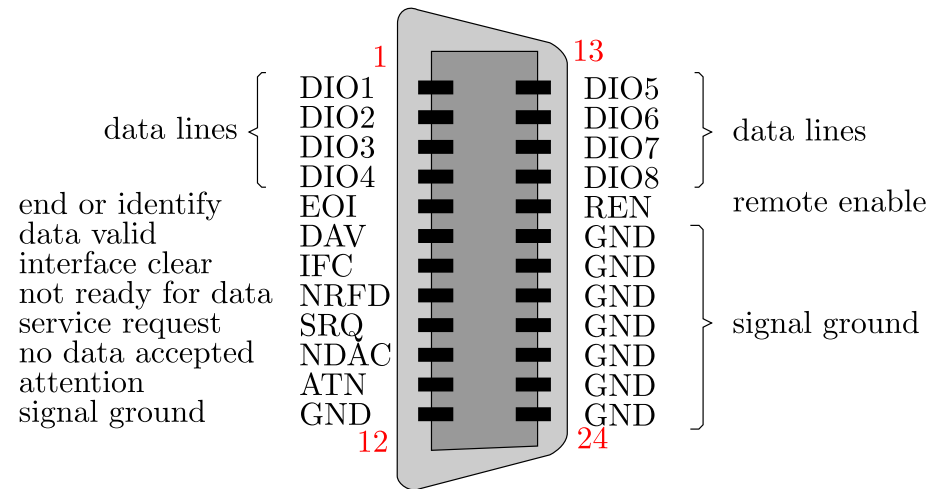


Abb. 12.34: Pin-Belegung im GPIB-Stecker.

Tab. 12.7: Belegung der Pins bei einer Schnittstelle nach der Norm IEEE 488.

12.8 Universal serial bus (USB)

- wird immer beliebter
- „hot plugging“ möglich
- Anwendersicher (Sicherheit vor dem Anwender)

Generationen:

- USB 1.x
- USB 2.0, with multiple updates and additions
- USB 3.x

	1996 USB 1.0	2001 USB 2.x	2011 USB3.x	2014/17 USB 3.1 & 3.2
Rates	ca. 1 MB/s	60 MB/s	ca. 1 GB/s	
Standard	Type A Type B		Type A Type B	
Mini		Mini A Mini B Mini AB	out-dated out-dated	
Micro		Micro A Micro B Micro AB	Micro B Micro AB	
Full Duplex				Type C
Spannung (V)	5.0	5.0	5.0	5.0
Leistung (W)	0.5	2.5	4.5	4.5

Tab. 12.8: Entwicklung der USB-Schnittstellen.

In Vor-USB-Zeiten:

- notwendige Einstellungen durch den Anwender:
 - Interrupts
 - Geschwindigkeit
 - Input/Output-Adressen
 - ...
- heute: USB ist selbstkonfigurierend

Überblick zu USB (Stand 2018):

- ein Master, mehrere Slave-Clients
- maximal 127 Geräte
- maximale Gesamt-Kabellänge: 15 m
- bis zu 5 Repeater
- Übertragungsrate bei USB 3.2 bis zu 1 800 MB/s
- Inline-Power (5 V & max. 15 W bei Type C)
- Topologie: physikalischer Baum, logischer Stern

Einsatzgebiete von USB:

- viele Schnittstellen wurden durch USB abgelöst (RS-232, Parallel-Schnittstelle, PS/2-Schnittstelle für Maus & Tastatur, GPIB, ...)
- heute: USB nahezu überall im Privatbereich
- USB setzt sich auch bei Labortechnik durch

Kabel und Stecker:

- 4 Adern (bei Standardtypen):
 - VCC & GND zur Spannungsversorgung
 - D_+ und D_- für Daten, differentiell
- Wellenwiderstand $Z = 90 \Omega$

Pegel:

- für USB 1.1 und USB 2.0 ohne Hi-Speed
- Bereiche der Pegel:
 - Pegel für LOW V_{OL} : 0.0 V – 0.3 V
 - Pegel für High V_{OH} : 2.8 V – 3.6 V
- differentielle Pegel:
 - differentielle 0: $D_+ = V_{OH}$ und $D_- = V_{OL}$
 - differentielle 1: $D_+ = V_{OL}$ und $D_- = V_{OH}$
 - single ended: $D_+ = 0$ und $D_- = 0$ durch Pull-Down-Widerstände

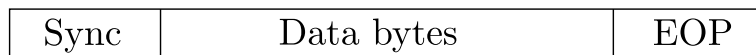


Abb. 12.35: Aufbau eines Datenpakets.

Single ended zero (SE0)		$D_+ = V_{OH}$ und $D_- = V_{OH}$
Data J state	low speed	differential 0
	full speed	differential 1
Data K state	low speed	differential 1
	full speed	differential 0
Idle state	low speed	$D_+ = V_{OH}$ und $D_- = V_{OL}$
	full speed	$D_+ = V_{OL}$ und $D_- = V_{OH}$
Resume state		Data K state
Start of packet (SOP)		transition from Idle to Data K state
End of packet (EOP)		SEO for 2 bit times then J state for 1 bit time
Disconnect & Reset		SEO for $> 2 \mu s$
Connect		Idle for $> 2 \mu s$

Tab. 12.9: Zustände und Signale des USB-Busses.

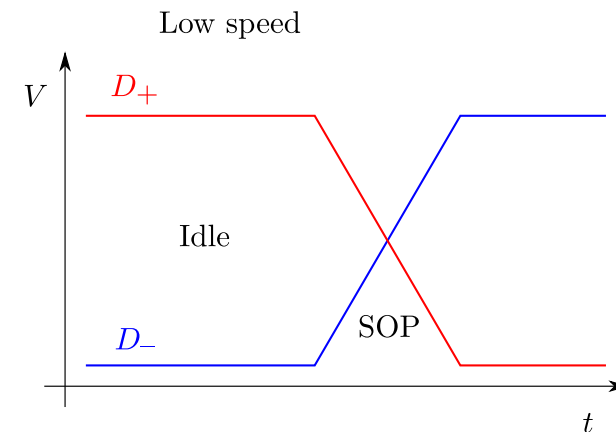


Abb. 12.36: Zustand „Idle“ und „Start of package“ bei Low-Speed.

Abschluss der Diskussion:

- „You need not be concerned with the detail of syncs, bit stuffing, and End Of Packet conditions, unless you are designing at the silicon level . . . “
- Es gibt kaum Möglichkeiten, eine USB-Schnittstelle außerhalb des vom Entwickler gesetzten Rahmens zu manipulieren.

Adapter: USB zu RS-232

- falls der PC keine RS-232 hat
- Vorsicht: funktioniert nicht so gut wie die Versprechen der meisten Hersteller



Abb. 12.37: Adapter von USB auf RS-232 von LogiLink.

Interface: USB zu SPI

- Ziel: selbstgebaute Elektronik mit dem PC verbinden
- ein komplettes Kit ist notwendig:
 - Hardware
 - Driver
 - sinnvoll: Software Beispiele
 - praktisch: Programmierung über VISA-Layer

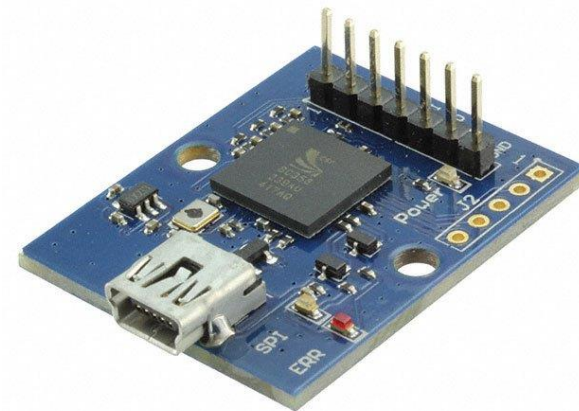


Abb. 12.38: Kit für die Umsetzung von USB auf SPI von Qualcomm.

12.9 Ethernet

Überblick:

- Familie von „computer networking technologies“
- benutzt in:
 - local area networks (LAN)
 - metropolitan area networks (MAN)
 - wide area networks (WAN)
- standardisiert mit IEEE 802.3 und Nachfolgern
- Verbindungsmedien:
 - „Twisted-pair“
 - Glasfasern
- Übertragungsraten je nach Typ: 1 Mbit/s – 100 Mbit/s

Kommunikationsinhalt:

- Unterteilung des Datenstroms in „Frames“
- die „Frames“ enthalten
 - Adresse des Senders
 - Adresse des Empfängers
 - Daten
 - Daten zur Fehlerkorrektur
- Fehlerkorrektur:
 - nicht durch Ethernet
 - sondern durch Protokolle wie TCP

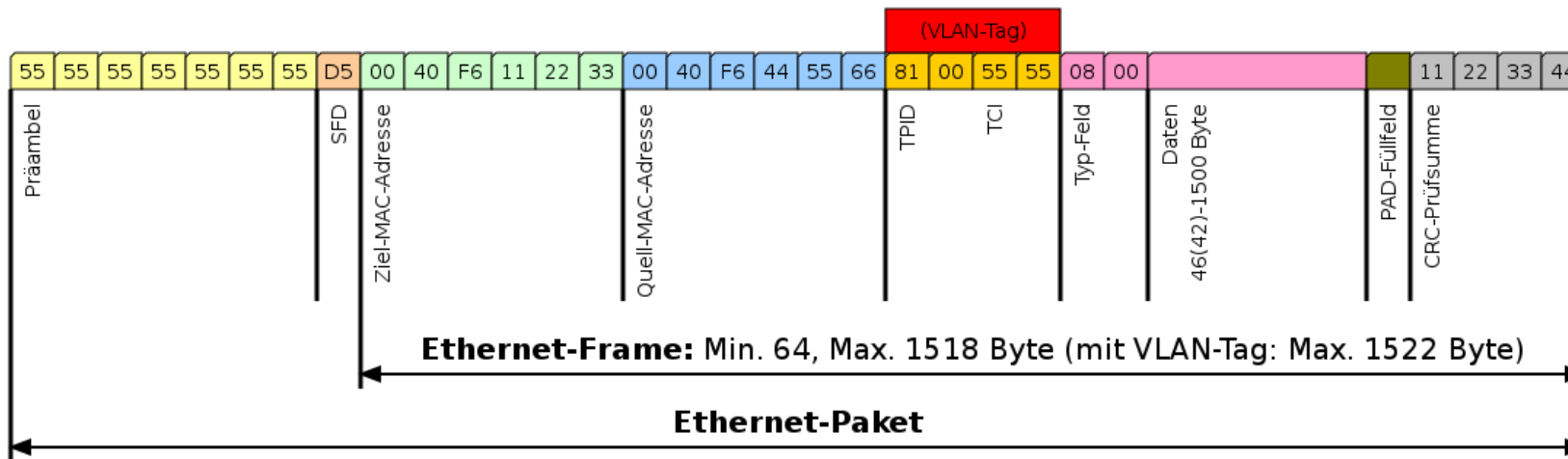


Abb. 12.39: Ethernet-Datenblockformat. Quelle: Wikipedia.

Kabel:

- achtadrig
- Stecker & Buchsen:
 - 8P8C-Modularstecker
 - anscheinend falsche Bezeichnung: RJ45
- Patches sind zu vermeiden
- PC an Switch: „straight through“
- PC an PC oder Switch an Switch: „crossed“



Abb. 12.40: 8P8C-Modularstecker und -buchse. Quelle: Wikipedia.

Transmission Control Protocol:

- kurz: TCP (aber nicht TCP/IP)
- verbreitetes Protokoll
- Vollduplex
- automatische Erkennung von Datenverlusten und deren Behebung
- fast der gesamte Internet- und Emailverkehr läuft über TCP
- Verbindung zwischen zwei Endpunkten
- benötigt Sockets, ein Paar aus
 - IP-Adresse
 - und Port
- zwei Paare (Quadrupel) charakterisieren eine Verbindung
- Beispiel für zwei Verbindungen an einem Rechner:
 - Lokaler Rechner, Port x, ferner Rechner X, Port y
 - Lokaler Rechner, Port x, ferner Rechner Y, Port z
- einfache Handhabung über Programmbibliotheken:
 - Windows: winsock.dll
 - Unix: bereits im Betriebssystem enthalten

12.10 Labortechnik

12.10.1 PC/PC-Kopplung via Ethernet

Ziel:

- Schnelle TCP-Verbindung zwischen zwei PCs
- marginaler Aufwand (< 4 h)
- selbstgeschriebenes Kommunikationsprotokoll

Winsock.dll:

- Windows-Socket-Programmierung
- Quelle: <https://msdn.microsoft.com>

Benötigtes Kabel: „crossed“

IP-Adresse des PCs finden (Windows):

- Start -> Alle Programme -> Zubehör -> Eingabeaufforderung
- Eingabe: ipconfig

Port:

- nahezu egal

Server-Client-Verbindung mit Code in VB6:

- tcpClient.Connect MyServerIP, MyServerPort
 - MyServerIP: IP-Adresse, z.B. 192.168.0.13
 - MyServerPort: z.B. 1111
- tcpClient.SendData MyText
 - sendet den String MyText
- tcpClient_DataArrival
 - ist eine Sub-Routine
 - ruft sich selbst auf, sobald Daten eingeht
 - innerhalb der Sub sollten die Daten extrahiert werden (z.B. mit tcpClient.GetData)
- tcpClient.GetData ReceivedText
 - schreibt eingegangene Daten in ReceivedText
- tcpClient.Close
 - schließt die Verbindung
- ...
- ...
- tcpServer.LocalPort = MyPort
- tcpServer.Listen
- tcpserver_ConnectionRequest ..
..(ByVal requestID As Long)
- ...

12.10.2 Auffinden von Schnittstellen & Geräten

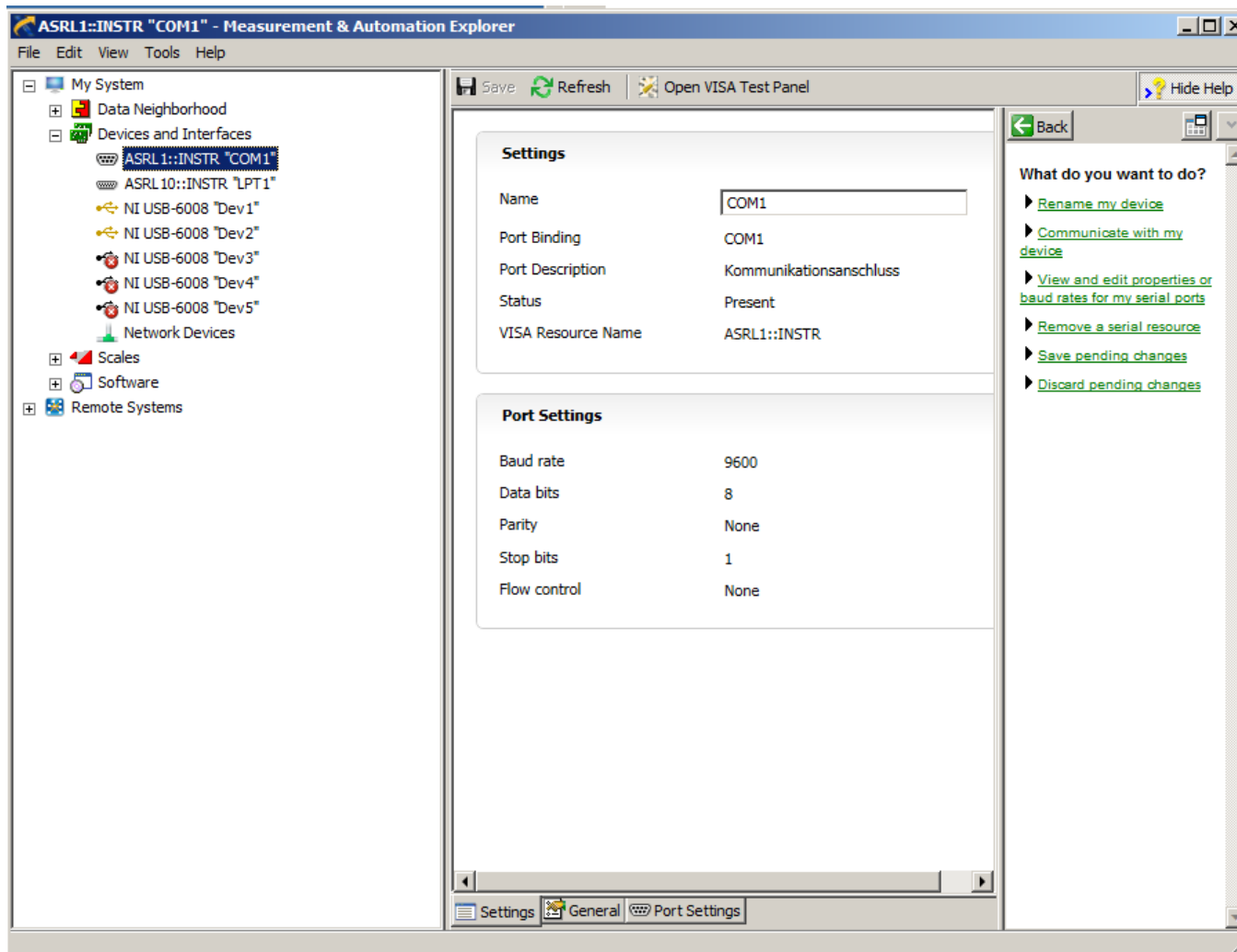


Abb. 12.41: Das Programm „NI Max“ von National Instruments.

Übliche Fragen:

- Sieht der PC das Peripheriegerät?
- Welche Adresse hat das Peripheriegerät?
- Können Daten übertragen werden?
- Welches Datenformat wird benötigt?

Literatur

- [1] U. Tietze, C. Schenk, and E. Gamm, *Halbleiter-Schaltungstechnik*, Springer, 2006.
- [2] P. Horowitz and W. Hill, *The Art of Electronics*, Cambridge University Press, 2015.
- [3] E. Hering, K. Bressler, and J. Gutekunst, *Elektronik für Ingenieure und Naturwissenschaftler*, Springer Vieweg, 2014.
- [4] Texas Instruments, Logic guide, 2017.
- [5] OnSemiconductor, LS TTL Data, 2000.
- [6] S. Sze, *Semiconductor Devices*, John Wiley & Sons, 1985.
- [7] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, *Introduction to Flash Memory*, Proc. IEEE **91**, 489 (2003).
- [8] K. Fricke, *Digitaltechnik*, Springer Vieweg, 2018.
- [9] R. Sbiaa, H. Meng, and S. N. Piramanayagam, *Materials with perpendicular magnetic anisotropy for magnetic random access memory*, phys. stat sol. RRL **5**, 413 (2011).