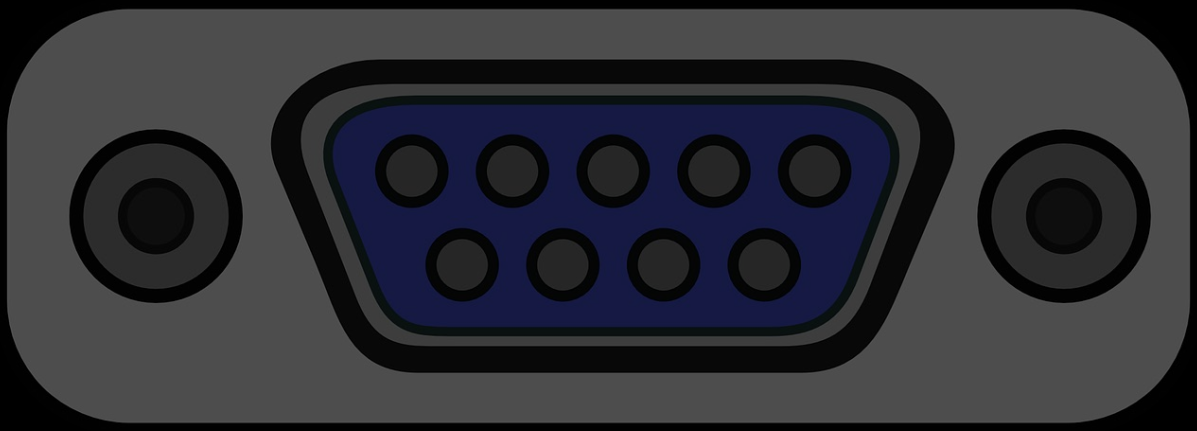


Serielle Schnittstellen

Kampl Maximilian

Inhaltsverzeichnis

SPI

Beschreibung

Steht für Serial Peripheral Interface und wurde im Jahre 1987 wurde von Susan C. Hill, sowie anderen, unter Leitung Motorolas entwickelt.

Das Bus-System stellt einen „lockeren“ Standard dar, mit dem digitale Schaltungen nach dem Master-Slave Prinzip verbunden werden.

Leitungen / Pins

Es gibt drei Leitungen, an denen jeder Teilnehmer angeschlossen ist:

- SLK – Serial Clock, welches vom Controller zur Synchro ausgegeben wird
- POCI – Peripheral Out / Controller in (Auch Master Input / Slave Output)
- PICO – Peripheral In / Controller out (Auch Master Output / Slave Input)

Außerdem gibt es noch die sogenannten SSL- Slave Select Leitungen und CSL – Chip Select Leitungen.

Da Master / Slave ein relativ problematischer Name ist gibt es Bemühungen die Pins umzubenennen

- SDO -Serial Data out
- SDI – Serial Data in

Schaltung (Kommunikation)

Bei der Kommunikation von SPIs gibt es keine Sender & Empfänger, man spricht hier eher von einem Austausch. Grund dafür ist, dass sowohl die Peripherie als auch der Controller jeweils 1 Bit gleichzeitig bekommen.

Jedoch geht jegliche Kommunikation im von der Peripherie aus, welche die SCK Impulse generiert. Und der Controller MUSS das Signal annehmen und verwerten, selbst wenn noch kein Ergebnis daliegt, misst die Peripherie die Polarität der PICO Leitung und bestimmt sich daraus das nächste Bit

Timing Diagramm

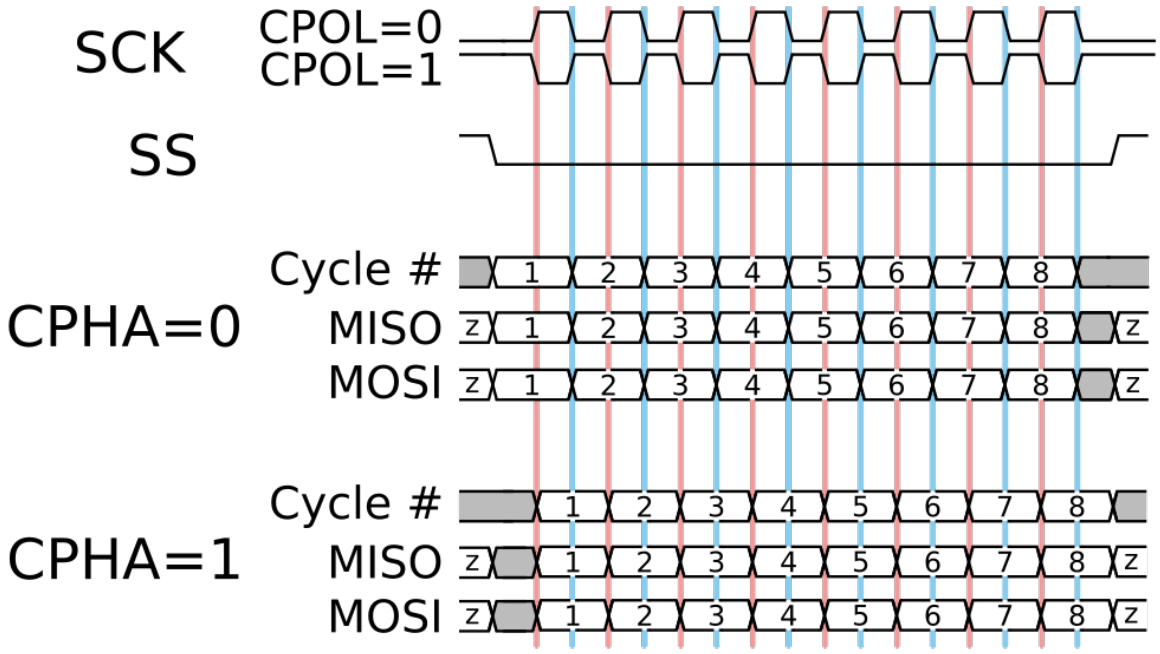
Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

CPOL (Clock Polarity)

- 0: Takt ist in Ruhe LOW, ein Wechsel auf HIGH zählt als steigende Taktflanke
- 1: Takt ist invertiert: in Ruhe HIGH, ein Wechsel auf LOW zählt als steigende Taktflanke

CPHA (Clock Phase)

- 0: Daten werden bei steigender Taktflanke (=abh. von CPOL) eingelesen, bei fallender ausgegeben
- 1: Daten werden bei fallender Taktflanke eingelesen, bei steigender ausgegeben



I2C

Beschreibung

Ebenso wie die SPIs sind auch ein I2Cs ein Datenbus. Die Abkürzung steht für Inter-Integrated Circuit und sie wurden 1982 von Philips entwickelt.

Der I2C wird hauptsächlich in Geräten für die Kommunikation von Schaltungsteilen verwendet.

Leitungen / Pins

- SCL/SLK
- SDA

Schaltung (Kommunikation)

In einem I2C gibt es mindestens einen Master und bis zu 127 Slaves. Ein Bus mit mehr als einem Master wird Multi-Master-Bus genannt.

Jeder Slave braucht seine eigene Adresse (7 o. 10 bit), damit sie zu jedem Master individuell sprechen können. Außerdem existiert auch noch eine Broadcastadresse, über welche die Slaves gleichzeitig angesprochen werden können.

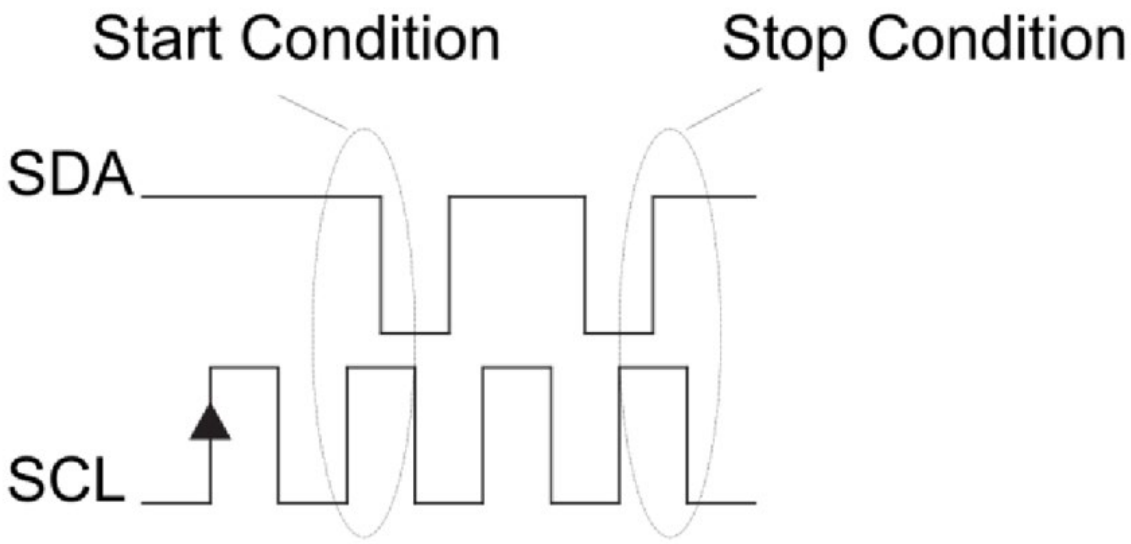
Gleich wie bei SPI kommt es nur zur Kommunikation, wenn der Master die Slaves anspricht. Jedoch anders wie bei dem SPI-Bussystem, wird hier festgelegt, ob empfangen oder gesendet wird.

Das Senden erfolgt durch Start/Stopp Bedingungen (Takt & Datenleitungszustände) von Seite des Masters aus und nach einer erfolgreichen Kommunikation senden die Slaves ihre Adresse und, im Schreibfall, ein Acknowledge.

Timing Diagramm

- Standardmodus: 100 kbit/s
- Fast mode: 400 kbit/s
- Fast mode+: 1 mbit/s
- High Speed mode: 3,4 mbit/s

Aufgrund von Pausen und asynchronem Betrieb ist die reale maximale Datenrate meist etwas niedriger. Wenn die Taktrate für einen Slave zu hoch ist, kann er durch "Clock Stretching" die Übertragung verlangsamen, indem er die Clock-Leitung auf GND zieht. Dies ist auf Bit- und Byte-Ebene möglich, jedoch nicht im High-Speed Mode auf Bit-Ebene.



UART

Beschreibung

UART steht für Universal Asynchronous Receiver Transmitter und ist eine serielle Schnittstelle, die asynchron arbeitet, also ohne festen Taktgeber. Sie wurde entwickelt, um die Kommunikation zwischen digitalen Geräten zu ermöglichen, und wird häufig in Mikrocontrollern, Computern und anderen elektronischen Geräten verwendet. Im Gegensatz zu synchronen Schnittstellen wie SPI oder I2C benötigt UART keine zusätzliche Taktleitung, sondern synchronisiert sich durch Start- und Stop-Bits. Der Datenaustausch erfolgt über zwei Leitungen, wobei ein Gerät als Sender und das andere als Empfänger fungiert.

Leitungen / Pins

UART verwendet in der Regel zwei Hauptleitungen:

- **TX** (Transmit): Diese Leitung dient zum Senden von Daten.
- **RX** (Receive): Diese Leitung dient zum Empfangen von Daten.

Manchmal gibt es zusätzliche Leitungen für die Handshaking-Methoden (Hardware-Flow-Control):

- **RTS** (Request to Send): Wird vom Sender verwendet, um anzuzeigen, dass er bereit ist, Daten zu senden.
- **CTS** (Clear to Send): Wird vom Empfänger verwendet, um dem Sender mitzuteilen, dass er bereit ist, Daten zu empfangen.

Andere Signale wie **DSR**, **DTR**, **RI**, und **DCD** können ebenfalls verwendet werden, sind jedoch weniger gebräuchlich.

Schaltung (Kommunikation)

Bei UART erfolgt die Kommunikation zwischen zwei Geräten asynchron, d.h. ohne gemeinsamen Takt. Daten werden in Form von Frames übertragen, die aus einem Start-Bit, den eigentlichen Datenbits (meist 8), einem optionalen Paritätsbit und einem oder mehreren Stop-Bits bestehen.

1. **Start-Bit**: Der Kommunikationsrahmen beginnt mit einem Start-Bit, das von der TX-Leitung als LOW gesendet wird, um den Empfänger zu informieren, dass eine Datenübertragung beginnt.
2. **Datenbits**: Nach dem Start-Bit folgen die Datenbits, die in der Regel 5 bis 9 Bits umfassen. Die Anzahl wird zuvor zwischen den Geräten festgelegt.
3. **Paritätsbit** (optional): Dieses Bit dient zur Fehlererkennung. Bei der geraden Parität (even parity) wird das Bit so gesetzt, dass die Gesamtanzahl der 1-Bits gerade ist, bei der ungeraden Parität (odd parity) ungerade.
4. **Stop-Bit(s)**: Das Stop-Bit beendet den Datenrahmen. Es kann ein oder zwei Stop-Bits geben, wobei die TX-Leitung auf HIGH gesetzt wird.

Wichtig ist, dass Sender und Empfänger vorab die gleichen Parameter wie Baudrate, Anzahl der Datenbits und Parität festlegen, um korrekt kommunizieren zu können.

Timing Diagramm

UART verwendet keinen festen Takt, sondern synchronisiert den Datenstrom über die Start- und Stop-Bits. Die Geschwindigkeit der Übertragung wird durch die **Baudrate** bestimmt, die die Anzahl der Symbole pro Sekunde festlegt. Typische Baudraten sind:

- 9600 Baud
- 19200 Baud
- 38400 Baud
- 115200 Baud

Die Datenübertragung erfolgt bei UART asynchron, d.h. es gibt keine feste Taktfrequenz, wie bei SPI oder I2C. Stattdessen wird die Baudrate im Voraus zwischen den beiden Geräten vereinbart.