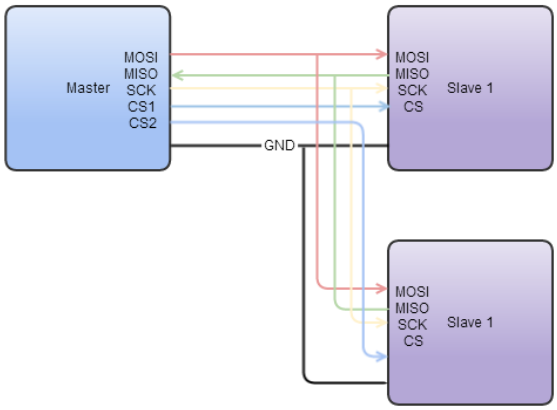
Serielle Schnittstellen

Bei einer seriellen Schnittstelle erfolgt die Übertragung Bit für Bit nacheinander, also seriell, in der Regel über eine Ader oder ein Adernpaar. Binäre Datenworte müssen also erst in einen seriellen Datenstrom umgewandelt werden.
Unterscheiden muss man Punkt-zu-Punkt-Verbindungen zwischen zwei Endpunkten oder -systemen, sowie seriellen Bussystemen mit mehr als zwei Teilnehmern.
Serielle Schnittstellen erweitern die Einsatzmöglichkeiten von Computern, Mikrocontrollern und Peripheriegeräten und schaffen Verbindungen zu anderen Systemen.

Serial Peripheral Interface

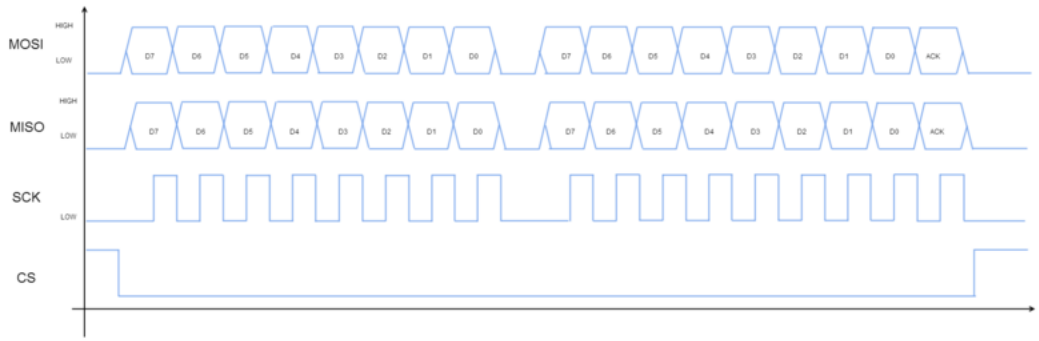
Das **S**erial **P**eripheral **I**nterface, kurz SPI oder auch Microwire genannt, ist ein Bussystem bestehend aus drei Leitungen für eine serielle synchrone Datenübertragung zwischen verschiedenen ICs.



Der Bus besteht aus folgenden Leitungen

- MOSI (**M**aster **O**ut -> **S**lave **I**n) auch SDO (**S**erial **D**ata **O**ut) oder DO
- MISO (**M**aster **I**n <- **S**lave **O**ut) auch SDI (**S**erial **D**ata **I**n) oder DI
- SCK (Serial Clock) - Schiebetakt

Zusätzlich zu diesen drei Leitungen wird für jeden Slave eine Slave Select (SS) oder auch Chip Select (CS) genannte Leitung benötigt, durch die der Master den Slave zur aktuellen Kommunikation selektiert. Dies geschieht dadurch, dass der Master die SS/CS-Leitung von High nach Low zieht. Oft ist mit dieser Aktivierung durch den Master auch eine Benachrichtigung für den Slave verbunden mit der ihm mitgeteilt wird, dass jetzt eine Nachricht beginnt, das nächste Byte also zum Beispiel als Kommando aufzufassen ist.



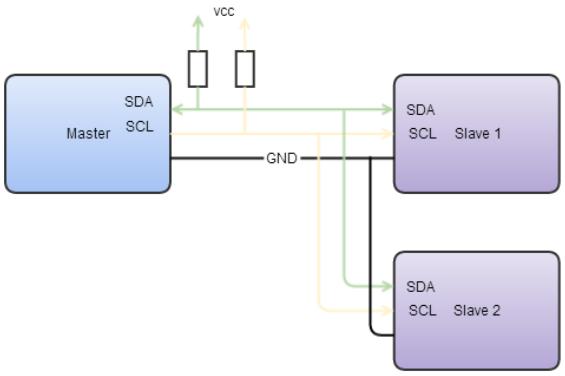
Die Übertragung geschieht so, dass der Master seine Datenleitung (MOSI) auf den Pegel des nächsten Bits bringt und dann an der SCK Leitung einen Puls ausgibt.

Gleichzeitig wird vom Master der Pegel an der Datenleitung vom Slave zum Master überwacht und ihr Zustand als nächstes einzulesendes Bit aufgefasst. Üblicherweise gibt es zumindest beim Master mehrere Einstellungen, die festlegen, welches der Grundzustand dieser SCK Leitung sein soll und welche Flanke des Taktes zur Datenübernahme herzunehmen ist (die steigende oder die fallende). Bei einigen Slaves ist diese Einstellung ebenfalls möglich, oft ist es aber so, dass per SPI anzusprechende IC eine feste Einstellung benutzen, an die sich der Master anpassen muss.

Für den SPI-Bus gibt es kein festgelegtes Protokoll. Die Taktpolarität (CPOL) und Phase (CPHA) können ebenfalls von Slave zu Slave unterschiedlich sein. Der SPI-Bus kann mit einer Taktfrequenz von vielen Megahertz betrieben werden. Es gibt viele verschiedene ICs die als Slave an dem SPI-Bus betrieben werden können.

I²C

I²C (gesprochen "I quadrat C") ist ein synchroner serieller Zweidraht-Bus, der jeweils eine bidirektionale (gemeinsamte) Daten- und Taktleitung verwendet. Die Bezeichnung steht für IIC, Inter-Integrated Circuit.



I²C wird auch als **TWI**, two wire interface (siehe AVR TWI) bezeichnet. Im PC wird ein dem I²C-Bus sehr ähnliches System benutzt, um z. B. die Daten eines SDRAM-Modules auszulesen. Dieser nennt sich SMBus (System Management Bus).

In einem I²C-Bus gibt es mindestens einen Master sowie bis zu 127 Slaves. Ein I²C-Bus mit mehreren Mastern wird als "Multi-Master-Bus" bezeichnet.

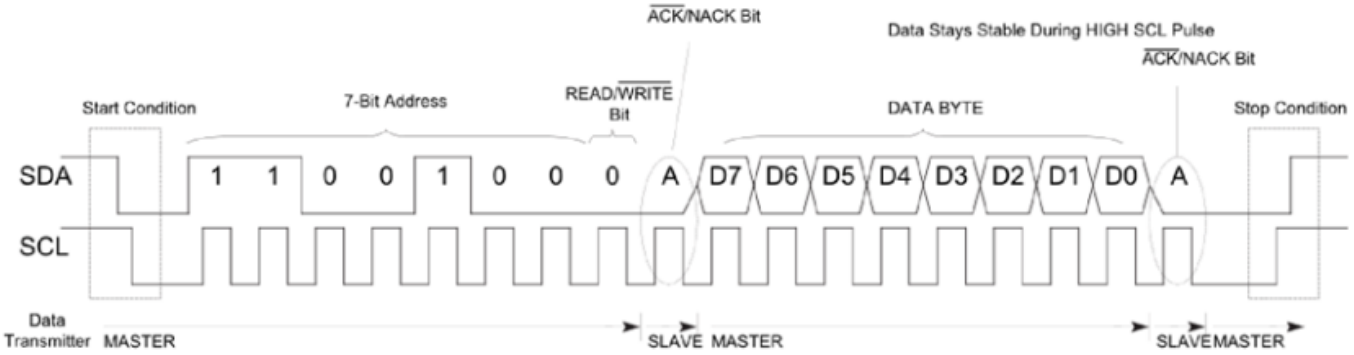
Die Slaves in einem Verbund müssen alle mit einer eigenen, individuellen Adresse codiert werden. Somit sind sie für jedem Master individuell anzusprechen. Darüber hinaus existiert ein sogenannter Broadcastkanal, mit dem alle Slaves gleichzeitig angesprochen werden können.

Ein Problem mit manchen Slaves ist der eingeschränkte Adressbereich, d.h. sie liegen auf einem definierten Bereich und haben darin z.b. nur 3 Bit als Programmieroption. Somit schließen diese Slaves indirekt andere Adressen aus.

Der (oder die) Master sprechen die Slaves jeweils aktiv an. Slaves können **nie** selbstständig beginnen, Daten zu senden. Um eine Kommunikation zu etablieren, übernimmt der Master, der Daten senden oder empfangen möchte, den Bus und gibt die (7-bit- bzw. 10-bit-)Adresse des Slaves aus, mit dem er kommunizieren möchte. Nach der Adresse teilt der Master dem entsprechenden Slave mit, ob er Daten senden oder empfangen möchte. Danach werden die

eigentlichen Daten (entweder vom Master oder Slave) auf den Bus gelegt. Hat der Master den Lese- oder Schreibvorgang abgeschlossen, so gibt er den Bus wieder frei. Sofern mehrere Master vorhanden sind, stellt ein logisches Protokoll sicher, dass sich diese nicht gegenseitig stören. Im Fall einer broad cast Operation kann ein time slot System sicherstellen, dass nie mehr als ein slave gleichzeitig antwortet.

Die Kommunikation auf Bitebene erfolgt seitens des Masters durch Senden von Start / Stopp Bedingungen, die sich aus der Kombination der Zustände von Takt- und Datenleitung ergeben. Um die erfolgreiche Kommunikation zwischen Master und Slave sicherzustellen, senden Slaves - ausser im broadcast Betrieb - nach erfolgreicher Dekodierung ihrer Adresse sowie im Schreibfall ein Acknowledge, in dem sie die Datenleitung ziehen. Diese muss dazu vom Master rechtzeitig freigegeben worden sein und beobachtet werden. Umgekehrt quittiert der Master in ähnlicher Weise den Empfang eines Datenbytes vom Slave und signalisiert somit weitere Empfangsbereitschaft. Der Takt wird immer vom Master getrieben.



UART

UART ist die Abkürzung für **U**niversal **A**synchronous **R**eceiver **T**ransmitter.

Beim UART handelt es sich um eine asynchrone serielle Punkt-zu-Punkt-Verbindung. Die Besonderheit bei der asynchronen Betriebsweise besteht darin, dass der Sender dem Empfänger kein eigenes Taktsignalauf einer eigenen Steuerleitung überträgt. Stattdessen synchronisiert sich der Empfänger über die Länge des Rahmens, vermittelt durch die Vorderflanke des neuen Start-Bits nach dem letzten empfangenen Stopp-Bit, sowie die eingestellte Baudrate (welche in diesem Fall der Bitrate entspricht). Weil der Beginn einer Übertragung mit dem Start-Bit zu beliebigen Zeitpunkten erfolgen kann, wird diese serielle Schnittstelle als *asynchron* bezeichnet.

Um eine Synchronisation gewährleisten zu können, ist die Anzahl der übertragbaren Datenbits innerhalb eines Rahmens eingeschränkt. Würde mehr als ein Byte in einen Rahmen verpackt, könnte die Synchronisation verloren gehen, was zu Fehlinterpretationen des Datenstromes und somit zu einer fehlerhaften Übertragung führen könnte. Wenn in einer Sendepause keine Daten zu übertragen sind, so legt

der Sender die Leitung auf die Polarität des Stopp-Bits. Weil der Empfänger sich mit jedem übertragenen Rahmen neu synchronisiert, ist es nicht notwendig, dass zwischen den übertragenen Rahmen ein zeitlicher Zusammenhang besteht. Nur für die Dauer eines einzelnen Rahmens müssen Sender und Empfänger synchron arbeiten, nicht länger.