# UART Schnittstelle

Eine UART (Universal Asynchronous Receiver/Transmitter) Schnittstelle ist ein Kommunikationsprotokoll, das für die serielle Datenübertragung zwischen einem Mikrocontroller oder einem Computer und anderen Geräten verwendet wird.

Sie wird oft verwendet, um Daten zwischen verschiedenen elektronischen Geräten zu senden und zu empfangen.

Eine UART-Schnittstelle ermöglicht die asynchrone Kommunikation, was bedeutet, dass Sender und Empfänger nicht auf einen gemeinsamen Takt synchronisiert sein müssen. Stattdessen werden Daten in Form von Bytes übertragen, wobei jedes Byte durch Start- und Stop-Bits gekennzeichnet wird.

UART-Schnittstellen werden in verschiedenen Anwendungen eingesetzt, wie zum Beispiel:

1. **Serielle Kommunikation zwischen einem Mikrocontroller und einem Computer:** UART wird häufig verwendet, um Daten zwischen einem Mikrocontroller und einem Computer zu übertragen. Dies ermöglicht die Steuerung des Mikrocontrollers oder den Austausch von Daten mit anderen Anwendungen auf dem Computer.
2. **Kommunikation zwischen verschiedenen Mikrocontrollern:** UART-Schnittstellen ermöglichen die Übertragung von Daten zwischen verschiedenen Mikrocontrollern in einem System. Dies kann dazu dienen, Informationen auszutauschen, Zustände zu synchronisieren oder verschiedene Funktionen innerhalb des Systems zu koordinieren.
3. **Anbindung externer Peripheriegeräte:** UART kann verwendet werden, um eine Verbindung zu externen Geräten wie Sensoren, Displays, GPS-Modulen usw. herzustellen. Durch die serielle Kommunikation können Befehle gesendet und Daten empfangen werden, um mit diesen Geräten zu interagieren.

Die spezifischen technischen Details einer UART Schnittstelle sind vom Mikrocontroller oder Kommunikationsgerät abhängig, in dem sie implementiert ist.

Hier sind einige technische Details einer UART Schnittstelle:

1. **Baudrate:** Die Baudrate bestimmt die Geschwindigkeit der Datenübertragung und wird in Bits pro Sekunde (bps) gemessen. Die Baudrate wird von beiden Seiten der Kommunikation vereinbart und muss übereinstimmen, um Daten korrekt zu senden und zu empfangen.
2. **Datenbits:** Die Anzahl der Bits, die für die Übertragung jedes Zeichens verwendet werden. Übliche Werte sind 5, 6, 7 oder 8 Bits.
3. **Parity Bits:** Ein Paritätsbit kann zur Fehlererkennung verwendet werden. Es kann auf gerade Parität (even parity) oder ungerade Parität (odd parity) eingestellt werden. Das Paritätsbit wird meist zusammen mit den Datenbits übertragen und ermöglicht es dem Empfänger, Übertragungsfehler zu erkennen.
4. **Stopbits:** Stopbits werden am Ende jedes übertragenen Zeichens verwendet, um das Ende der Kommunikation anzuzeigen und den Empfänger bereit für das nächste Zeichen zu machen. Übliche Werte sind 1 oder 2 Stopbits.
5. **Hardware- oder Software-UART:** Eine UART Schnittstelle kann entweder als Hardwarekomponente auf dem Mikrocontroller oder als softwarebasierte Emulation implementiert sein. Hardware-UARTs bieten in der Regel eine bessere Leistung und Zuverlässigkeit, während Software-UARTs nur mittels Softwarelogik realisiert werden und auf zusätzliche Ressourcen des Mikrocontrollers angewiesen sind.

6. **Puffergröße:** Eine UART Schnittstelle enthält oft einen Datenpuffer, der empfangene oder zu sendende Daten zwischenspeichert. Die Größe dieses Puffers kann je nach Implementierung und Anforderungen variieren.
7. **Voll- oder Halbduplex:** Eine Voll-Duplex-UART kann gleichzeitig senden und empfangen, während eine Halbduplex-UART nur in eine Richtung gleichzeitig arbeiten kann. Bei der Halbduplex-UART müssen Sende- und Empfangsrichtung wechseln.
8. **Handshake-Unterstützung:** Einige UART Schnittstellen bieten Handshake-Funktionen wie RTS (Request-to-Send) und CTS (Clear-to-Send). Diese Signale ermöglichen eine Steuerung des Datenflusses zwischen Sender und Empfänger und verhindern Datenverlust in Situationen, in denen der Empfänger noch nicht bereit ist, Daten zu empfangen.

*Dieser Inhalt wurde durch eine K.I. erstellt.*