

Technische Informatik



Schaltkreistechnik und Interfacing

- **Definition Interface**
- **Arten der seriellen Kommunikation**
- **Serielle Schnittstelle (RS232 / V.24 / COM)**
- **I2C-Bus**

Literatur

- Elektronik 4: Digitaltechnik, K. Beuth, Vogel Fachbuch
- Digitaltechnik, K. Fricke, Springer Vieweg
- Digitaltechnik, R. Voitowitz, Springer
- Grundlagen der Digitaltechnik, G. W. Wöstenkühler, Hanser



Definition Interface

Was ist Interface?

- Interface ist der englische Begriff für Schnittstelle zwischen verschiedenen Komponenten eines Kommunikationssystems, über die die Informationsverarbeitung oder der Informationsaustausch stattfindet.
- Eine Schnittstelle verbindet Systeme mit unterschiedlichen mechanischen, elektrischen und physikalischen Eigenschaften.
- Die Informationen können in unterschiedlichen Formen ausgetauscht werden zum Beispiel in Form von logischen Größen, physikalischen Größen, digital oder analog.

Es wird zwischen folgenden Schnittstellen unterschieden:

- **Common interface:** allgemeine Schnittstelle (zum Beispiel für Digitalfernsehen)
- **Software interface:** Softwareschnittstelle (Kommunikations- und Programmierschnittstellen)
- **Network interface:** Netzwerkschnittstelle (zwischen Netzwerkkomponenten)
- **Data interface:** Datenschnittstelle (zwischen Programmen)
- **User interface:** Benutzerschnittstelle (Menschen agieren beispielsweise über Anzeigen, Bedienelemente oder grafische Benutzeroberflächen mit einer Software oder einem Gerät.)

- Hardware interface: Hardwareschnittstelle (wichtig für die Kompatibilität zwischen Hardware-Komponenten, zum Beispiel AGP, PCI-Bus, USB oder FireWire)
- Maschinenschnittstelle (zwischen physischen Systemen)
- Nicht als *Interface* bezeichnet werden allgemeine mechanische Verbindungselemente der Konstruktionslehre, die der Informationsübermittlung dienen (etwa Seilzüge oder Federn).

:

- Überall dort, wo unterschiedliche Systeme miteinander verbunden werden müssen, befinden sich Schnittstellen.
- Sie bilden den Übergang von einem System in ein anderes und können zum Datenaustausch oder zur Kommunikation verwendet werden.
- Gemeinsame Eigenschaften definiert die Spezifikation einer Schnittstelle.
- Zur Spezifikation gehört ein Protokoll für den Datenaustausch und die Kommunikation.
- Definierte Schnittstellen sind in elektronischen Systemen eine Voraussetzung für die Interoperabilität mit anderen Baugruppen, Geräten und Systemen.

- Eine Schnittstelle wird durch eine Menge von Regeln beschrieben, der *Schnittstellenbeschreibung*.
- Neben der Beschreibung, welche Funktionen vorhanden sind und wie sie benutzt werden, gehört zu der Schnittstellenbeschreibung auch ein sogenannter *Kontrakt*, der die Semantik der einzelnen Funktionen beschreibt.
- Standardisierte Schnittstellen bieten den Vorteil, dass Komponenten oder Module, die die gleiche Schnittstelle unterstützen, gegeneinander ausgetauscht werden können, das heißt, sie sind miteinander kompatibel.
- Es kommt häufig vor, dass zwei Teilnehmer der Kommunikation unterschiedliche, aber zueinander passende Schnittstellen besitzen müssen (Kompatibilität, z. B. Stecker – Buchse).



Arten der seriellen Kommunikation

Welche Arten der seriellen Kommunikation gibt es?

- Die serielle Kommunikation ist eine häufig verwendete Methode zum Datenaustausch zwischen Computern und Peripheriegeräten.
- Die serielle Übertragung zwischen Sender und Empfänger unterliegt strengen Protokollen, die für Sicherheit und Zuverlässigkeit sorgen und zu ihrer Langlebigkeit geführt haben
- Diese Protokolle unterstützen die asynchrone Datenübertragung und die synchrone Datenübertragung.
- Die serielle Kommunikation nutzt eine serielle digitale Binärmethode für den Datenaustausch.
- Sie verwendet eine Reihe von Schnittstellen und seriellen Kommunikationsprotokollen, darunter RS232, RS485, SPI und I2C.

Asynchrone Datenübertragung

- Bei Verwendung eines asynchronen seriellen Protokolls muss der Absender vor der Übertragung von Daten ein Synchronisationssignal bereitstellen.
- Dies muss vor jeder übertragenen Nachricht erfolgen.
- Die Merkmale eines asynchronen Protokolls umfassen:
 - Zwischen dem Absender und dem Empfänger der Nachricht ist kein Taktsignal erforderlich.
 - Datenübertragung über größere Entfernungen wird unterstützt.
 - Die Zuverlässigkeit wird durch asynchrone Datenübertragung erhöht.

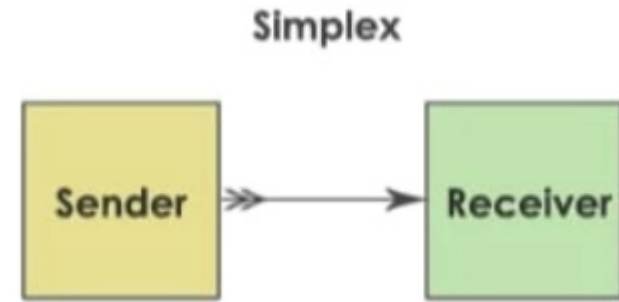
Synchrone Datenübertragung

- Bei Verwendung des synchronen seriellen Protokolls verwenden Sender und Empfänger während der Datenübertragung dasselbe Taktsignal.
- Einige andere Merkmale der synchronen Übertragung sind:
 - Synchrone Protokolle unterstützen höhere Datenübertragungsraten.
 - Das Taktsignal muss zwischen Sender und Empfänger kommuniziert werden.
 - Eine Master / Slave-Konfiguration ist erforderlich, um die synchrone Datenübertragung zu implementieren.

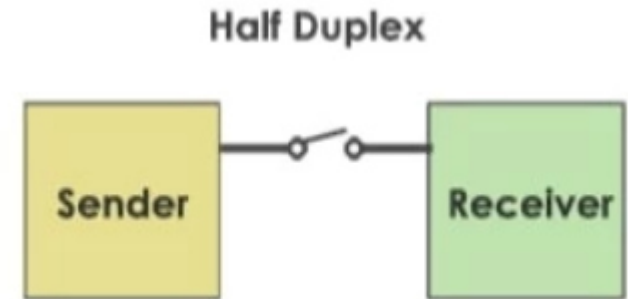
Asynchrone Kommunikationsprotokolle

- Im Folgenden sind einige der am häufigsten verwendeten asynchronen Kommunikationsschnittstellen aufgeführt, auf die Sie stoßen werden.
- Dies sind Beispiele für verschiedene Arten von seriellen Protokollen, die alle eine asynchrone Datenübertragung verwenden, um die Kommunikation zwischen Geräten oder Anwendungen zu ermöglichen.

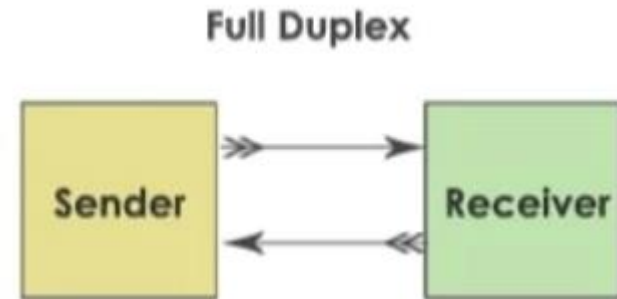
- **Die Simplex-Methode** implementiert die Datenübertragung in eine Richtung.
- In diesem Schema ist zu einem bestimmten Zeitpunkt nur die Quelle oder das Ziel aktiv.
- Wenn die Quelle Daten sendet, hat der Empfänger keine andere Wahl, als die Übertragung zu akzeptieren.
- Der Simplex-Modus wird zum Senden von Fernseh- oder Radiosignalen verwendet.



- **Halbduplex-Modus** ermöglicht, dass sowohl Quelle als auch Ziel aktiv sind, jedoch nicht gleichzeitig.
- Die Übertragung erfolgt jeweils nur in eine Richtung.
- Ein typisches Beispiel ist die Nutzung des Internets.
- Wenn Sie eine Webseite von Ihrem Computer aus anfordern, verarbeitet der Server die Anforderung.
- Dieser wird dann zum Absender, wenn die Informationen an Ihren Computer zurückgesendet werden, der jetzt der Empfänger ist.



- **Vollduplex-Modus** ist die weltweit am weitesten verbreitete Form der seriellen Kommunikation.
- Quelle und Ziel sind beide aktiv und können gleichzeitig Daten senden und empfangen.
- Ihr Smartphone ist ein hervorragendes Beispiel für den Vollduplex-Modus in Aktion.



Serielle Kommunikation vs. Parallele Kommunikation

- Die serielle Kommunikation sendet Daten bitweise. Folglich werden weniger E/A-Leitungen (Eingabe-Ausgabe-Leitungen) benötigt, um sie zu implementieren, als für die parallele Übertragung.
- Dies führt zu weniger Interferenzen und einer Verringerung des Platzbedarfs.
- Es reduziert auch die Kosten des eingebetteten Systems und ermöglicht die zuverlässige Übertragung von Daten über große Entfernungen.
- Datenkommunikationsgeräte (DCE) wie Modems nutzen die serielle Datenübertragung.

- Weitere E/A-Leitungen sind erforderlich, um die parallele Kommunikation zu implementieren.
- Wenn Daten in einem Block von 8, 16 oder 32 Bit gesendet werden, benötigt jedes Bit eine eigene physikalische E/A-Leitung.
- Die Geschwindigkeit der parallelen Übertragung ist schneller als die der seriellen Übertragung, erfordert jedoch eine größere Anzahl von E/A-Leitungen.
- Die parallele Datenübertragung wird in Personal Computern verwendet, um Daten mit internen Komponenten wie dem Direktzugriffsspeicher (RAM) oder der CPU auszutauschen.

Die Vor- und Nachteile der seriellen und parallelen Kommunikation werden in dieser Vergleichstabelle hervorgehoben.

Serieller Bus

Sendet bei jedem Takt ein Datenbit

Bessere Methode für die Fernkommunikation

Langsame Übertragungsgeschwindigkeit

Erfordert eine einzelne Leitung für die Datenübertragung

Niedrige Installationskosten

Beispiel: Computer zu Computer

Paralleler Bus

Überträgt einen Datenblock gleichzeitig

Wird hauptsächlich für die Kommunikation über kurze Entfernungen verwendet

Schnellere Kommunikation

Benötigt n Zeilen für die Übertragung von n Bits

Höhere Installationskosten

Beispiel: Computer zu Multifunktionsdrucker

Die Baudrate

- Die Baudrate bezieht sich auf die Übertragungsgeschwindigkeit zwischen Sender und Empfänger.
- Es wird in Bits pro Sekunde ausgedrückt. Einige häufig verwendete Baudraten sind 1200, 2400, 4800, 9600 und 57600.
- Sowohl Sender als auch Empfänger müssen auf die gleiche Baudrate eingestellt sein. In diesem Fall handelt es sich um Ihren Laptop und Ihr Mobilgerät.

Hinweis: Eine höhere Baudrate führt zu einer schnelleren Datenübertragung. Sie könnten erwägen, 115200 Baud als Grenze zu verwenden, um die Möglichkeit zu minimieren, dass der Empfänger nicht mit höheren Frequenzen umgehen kann.

Fazit

- Die serielle Kommunikation ist ein kritisches Konzept, das in vielen Bereichen der Elektronik und eingebetteten Systeme implementiert wird.
- Für Anwendungen muss ein gültiges serielles Protokoll ausgewählt werden, um einen ordnungsgemäßen Wechselkurs zu gewährleisten, wenn zwei Geräte denselben Bus verwenden.



Serielle Schnittstelle

RS-232

- **Recommended Standard 232** ist ein Standard für eine serielle Schnittstelle, der in den frühen 1960er Jahren vom US-amerikanischen Standardisierungsgremium Electronic Industries Association (EIA) erarbeitet wurde und bis in die 2010er Jahre häufig bei Computern vorhanden war.
- Die serielle Schnittstelle entspricht den Standards RS-232(C) und V.24.
- Die englische Bezeichnung COM-Port, abgeleitet von "Communication" (Kommunikation), wird allerdings am häufigsten verwendet.
- Mit COM1, COM2, COM3, usw. wird einer physischen existierenden Schnittstelle eine logische Bezeichnung durch das BIOS und das Betriebssystem zugeteilt.

Anwendung

- Mainframes und Text-Terminals wurden bis in die frühen 1990er-Jahre unter Zuhilfenahme von Modems durch Punkt-zu-Punkt-Verbindungen über die Telefonleitung zusammengeschlossen.
- Die Datenübertragung zwischen den beiden Systemen erfolgte seriell.
- Durch den ursprünglichen Verwendungszweck bedingt, weist die Schnittstelle einige Asymmetrien bei der Definition der Steuerleitungen auf, die bei den später üblich gewordenen Anwendungen in völlig anderen Bereichen zu Verschaltungsproblemen führen können.

Aktuelle Verwendung

- Weltweit werden immer weniger Geräte mit RS-232-Schnittstelle produziert. Beispiele sind Service- und Konfigurationsanschlüsse bei Geräten wie z. B. Router, Switches, Speichersysteme, Laborgeräte und Point-Of-Sale-Terminals.
- Alternative serielle Schnittstellen bieten zuverlässigere und schnellere Verbindungsmöglichkeiten. Nur noch wenige PCs werden mit einem COM-Port ausgeliefert, Notebookhersteller bieten diese Ausstattungsoption nahezu gar nicht mehr an.
- Um Geräte, die eine RS-232-Schnittstelle besitzen, mit Computern ohne diese betreiben und programmieren zu können, gibt es Konverter von USB auf RS-232.
- Auch Steckkarten mit RS-232-Schnittstellen für PCs werden angeboten.

Definition

- RS-232 definiert die Verbindung zwischen dem *Terminal* (Datenendeinrichtung (DEE), engl. data terminal equipment (DTE)) und dem *Modem* (Datenübertragungseinrichtung (DÜE), engl. data communication equipment (DCE)), was Timing und Spannungspegel betrifft.
- Als Steckverbinder wurden die 25-polige D-Sub empfohlen (nicht vorgeschrieben). Das Übertragungsprotokoll ist nicht Bestandteil des Standards.
- Allgemein sind die Parameter unter Serielle Datenübertragung erläutert

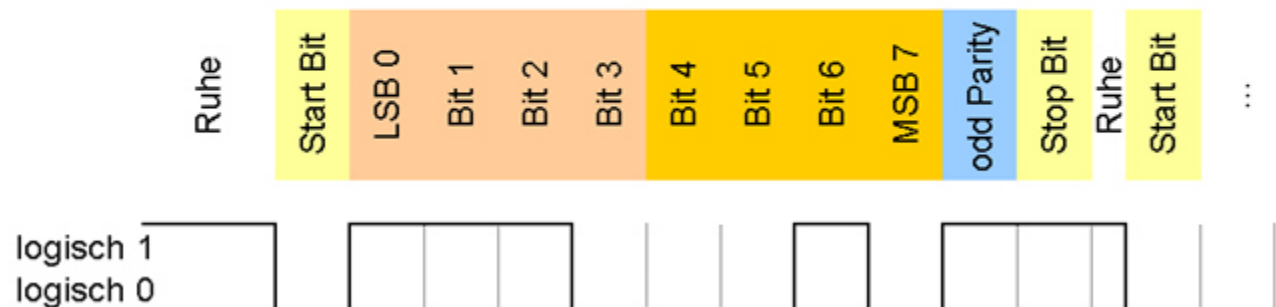
- Die Übertragung erfolgt in Wörtern.
- Ein Wort entspricht dabei je nach Konfiguration fünf bis neun Bits, in dem dann ein einzelnes Zeichen kodiert ist.
- Meistens erfolgt die Kodierung gemäß ASCII.
- Häufig kommen auch (ASCII-)SteuerCodes für die Ansteuerung eines Terminals wie VT100 zum Einsatz, diese sind im RS-232-Standard jedoch nicht definiert.
- Üblich ist daher, sieben bzw. acht Datenbits zu übertragen.
- Jedoch ist beispielsweise auch (nach Anpassung der Signalpegel) die Verarbeitung des 5-bit-Fernschreib-Codes möglich.

- Eine RS-232-Verbindung arbeitet (*bit-*)*seriell* mit je einer Datenleitung für beide Übertragungsrichtungen.
- Das heißt, die Bits werden nacheinander auf einer Leitung übertragen, im Gegensatz zur parallelen Datenübertragung.
- Die dafür nötige Seriell-Parallel-Wandlung geschieht meistens in sog. UARTs (entweder als integriertes Modul in einem Mikrocontroller oder als Einzelbaustein).
- Obwohl es zahllose andere serielle Schnittstellenarten gibt, wird die RS-232 traditionell „serielle Schnittstelle“ genannt, weil sie früher speziell im PC-Bereich die einzig übliche war.

- Die Datenübertragung erfolgt *asynchron*, es existiert also kein gemeinsamer Takt.
- Jeder Teilnehmer kann bei freier Leitung, zu jedem beliebigen Zeitpunkt, vollständige Datenwörter übertragen.
- Die Synchronisation in der Übertragung erfolgt durch den Empfänger als sogenannte Wortsynchronisation, also am Anfang durch die Signalflanke des Startbits.

Synchronisation
Daten low & high
Check

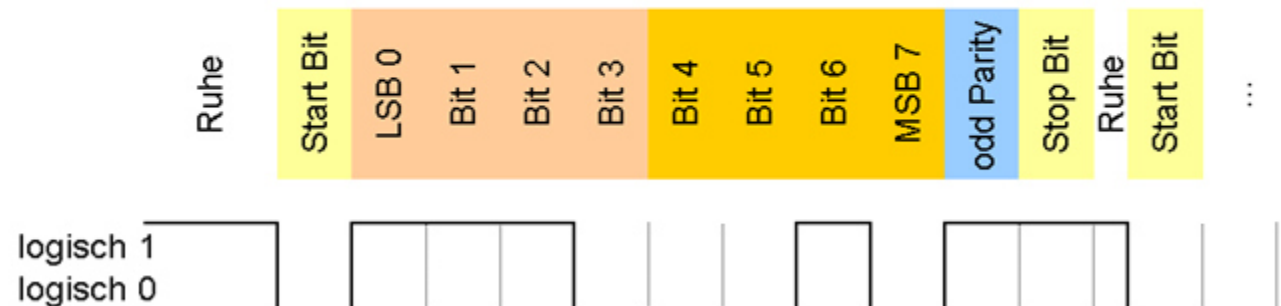
9600 8O1 = 9600 Baud; 8 Datenbits; odd Parity; 1 Stopbit
ASCII "G" = \$47 = 0100 0111



- Die Synchronisation des Empfängers geschieht mit dem Start der Übertragung auf der Datenleitung, da das Stopp-Bit bzw. der Ruhezustand auf der Leitung den inversen Pegel zum Start-Bit aufweist.
- Der Empfänger synchronisiert sich so in die Mitte der einzelnen Datenbits und tastet die folgenden Bits des Datenwortes mit seiner eigenen Bitrate ab.
- Damit das funktioniert, dürfen die Bitraten von Sender und Empfänger nur einige Prozent voneinander abweichen.

Synchronisation
Daten low & high
Check

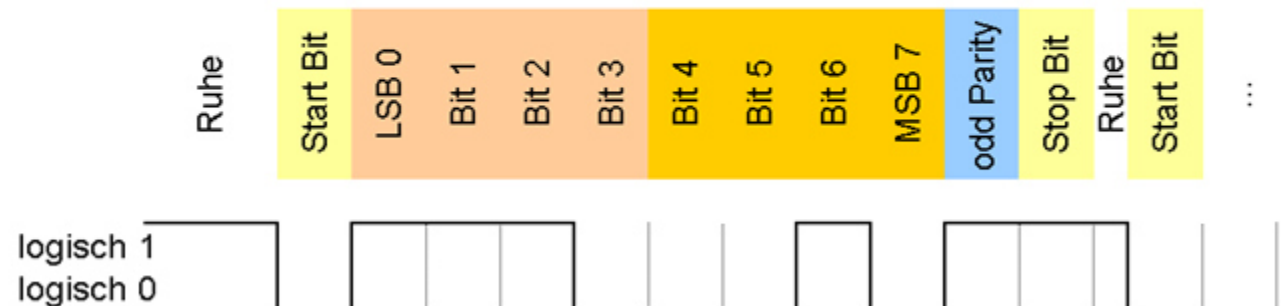
9600 8O1 = 9600 Baud; 8 Datenbits; odd Parity; 1 Stopbit
ASCII "G" = \$47 = 0100 0111



- Jedes übertragene Wort muss somit von einem Startbit (logischer Wert 0) eingeleitet und mit mindestens einem Stopp-Bit (logischer Wert 1) abgeschlossen werden.
- Das Stopp-Bit ist kein Bit im eigentlichen Sinne, sondern bezeichnet die Mindestlänge der Pause bzw. des Ruhezustands.
- Daher können zwischen zwei Wörtern beliebig viele Stopp-Bits vorliegen, auch nichtganzzahlige Werte wie 1,5 Stopp-Bits.

Synchronisation
Daten low & high
Check

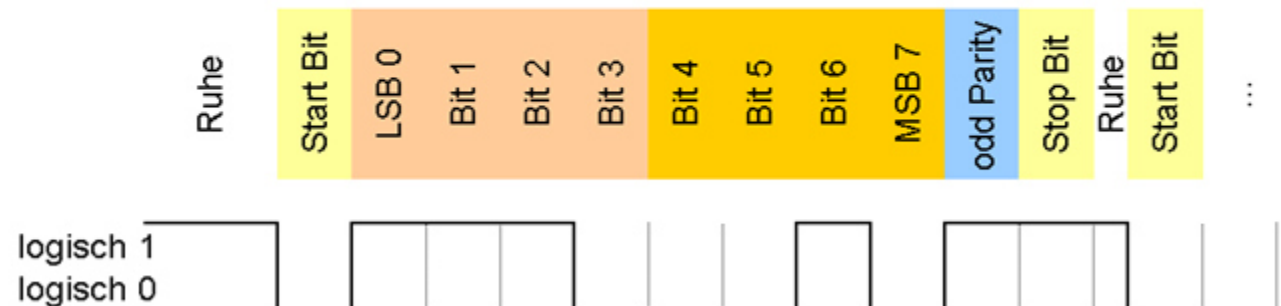
9600 8O1 = 9600 Baud; 8 Datenbits; odd Parity; 1 Stopbit
ASCII "G" = \$47 = 0100 0111



- Damit ist gemeint, dass die Mindestdauer der Pause der Zeitdauer von 1,5 Bitzellen entspricht.
- Der Grund liegt darin, dass manche UARTs zwischen dem Empfang zweier Wörter eine etwas längere Pause von mehr als einer Bitdauer benötigen.
- Zwischen Start- und Stopp-Bit(s) werden die eigentlichen Nutzdaten (Datenbits) über die Taktzeit unverändert (NRZ-codiert) übertragen.

Synchronisation
Daten low & high
Check

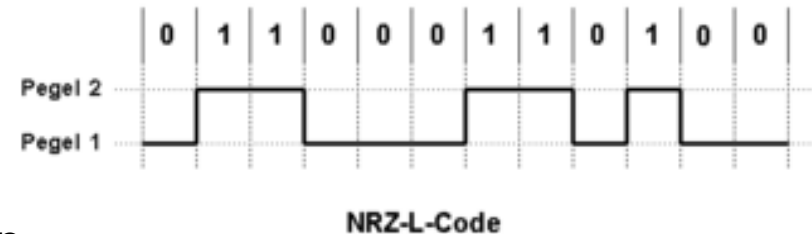
9600 8O1 = 9600 Baud; 8 Datenbits; odd Parity; 1 Stopbit
ASCII "G" = \$47 = 0100 0111



Non Return to Zero

- **Non-Return-to-Zero** und **Non-Return-to-Zero-Inverted**, abgekürzt **NRZ/NRZI**, vereinzelt auch als **Wechselschrift** bezeichnet, sind Begriffe aus der Digitaltechnik und beschreiben die einfachsten Leitungscodes für binäre Signale.
- Im Gegensatz zum RZ-Code bestehen die beiden binären Symbol aus konstanten Leitungszuständen (meist Spannungen).
- Die Bezeichnung Non-Return-to-Zero bezieht sich dabei nicht auf einen womöglich unzulässigen Spannungswert von 0 V, sondern darauf, dass es im Gegensatz zum RZ-Code keinen dritten Spannungswert gibt, der für einen Teil einer Symboldauer angelegt wird.
- Eine andere Auslegung besagt, dass die Spannung in der Mitte des Bits nie auf den Wert 0 zurückfallen kann.

- Der NRZ-Code ordnet direkt jedem Bit-Wert einen Leitungszustand zu.
- Er kann ohne weiteres verwendet werden, wenn in den Nutzdaten keine langen konstanten Folgen auftreten, wie etwa bei ASCII-kodierten Texten.
- Die Grenze für 'lang' kann recht kurz sein, etwa für ein Bandlaufwerk mit Gleichlaufschwankungen.
- Die NRZ-Kodierung ist im Allgemeinen auch nicht gleichanteilsfrei und damit insbesondere bei magnetischer Datenaufzeichnung problematisch.
- Eine einfache galvanische Trennung im Signalübertragungsweg mittels Impulstransformatoren ist daher auch nicht möglich.
- UARTs z. B. verwenden die NRZ-Kodierung.

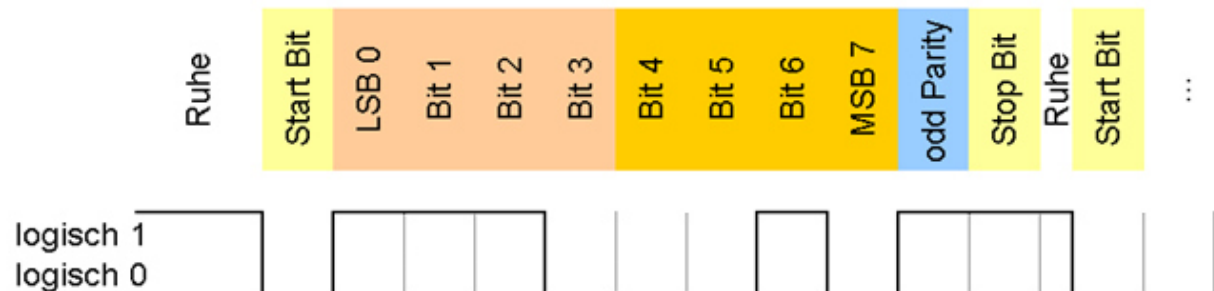


Paritätsbit

- Die Paritätskontrolle dient der Erkennung fehlerhaft übertragener Informationswörter.
- Als *Informationswort* wird hier eine Folge von Bits bezeichnet.
- Die „Parität“ bezeichnet die Anzahl der mit 1 belegten Bits im Informationswort und heißt gerade (engl. „even“), wenn die Anzahl dieser Bits gerade ist, andernfalls ungerade (engl. „odd“).

Synchronisation
Daten low & high
Check

9600 8O1 = 9600 Baud; 8 Datenbits; odd Parity; 1 Stopbit
ASCII "G" = \$47 = 0100 0111

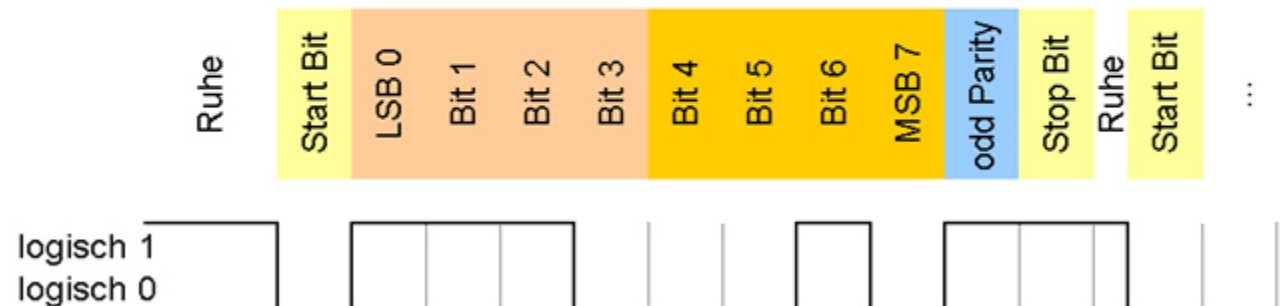


- Die Paritätskontroll-codierung hängt dem *Informationswort* ein *Paritätskontrollbit*, auch Paritybit genannt, an.

Summe der Einsen eines Informationsworts	Wert des Paritätsbits bei Even-Parity	Wert des Paritätsbits bei Odd-Parity
gerade	0	1
Ungerade	1	0

Synchronisation
Daten low & high
Check

9600 8O1 = 9600 Baud; 8 Datenbits; odd Parity; 1 Stopbit
ASCII "G" = \$47 = 0100 0111



Beispiel 1 (Even-Parity):

- Ist für die Datenübertragung Even-Parity (Paritätssumme gerade → Paritybit: 0, Paritätssumme ungerade → Paritybit: 1) festgelegt, so gilt für die beiden nachfolgenden Beispiele:
 - Das Informationswort 0011.1010 hat vier Einsen. Vier ist eine gerade Zahl, das Paritätskontrollbit ist also die Null, und das resultierende Codewort ist 0011.1010 **0**.
 - Das Informationswort 1010.0100 hat hingegen eine ungerade Paritätssumme und wird in das Codewort 1010.0100 **1** codiert.

Beispiel 2 (Odd-Parity):

- Ist für die Datenübertragung Odd-Parity (Paritätssumme gerade → Paritybit: 1, Paritätssumme ungerade → Paritybit: 0) festgelegt, so gilt für die beiden nachfolgenden Beispiele:
 - Das Informationswort 0011.1010 hat vier Einsen. Vier ist eine gerade Zahl, das Paritätskontrollbit ist also die Eins, und das resultierende Codewort ist 0011.1010 **1**.
 - Das Informationswort 1010.0100 hat hingegen eine ungerade Paritätssumme und wird in das Codewort 1010.0100 **0** codiert.

- Für die Datenleitungen (TxD und RxD) wird eine negative Logik verwendet, wobei eine Spannung zwischen -3 V und -15 V (ANSI/EIA/TIA-232-F-1997) eine logische Eins und eine Spannung zwischen $+3\text{ V}$ und $+15\text{ V}$ eine logische Null darstellt.
- Signalpegel zwischen -3 V und $+3\text{ V}$ gelten als undefiniert.
- In der sogenannten positiven Logik kodiert der *High*-Pegel den Binärwert 1 und der *Low*-Pegel den Binärwert 0, ***in negativer Logik stellt der High-Pegel die 0 und der Low-Pegel die 1 dar.***
- Bestimmte Anwendungen verwenden eine negative Logik.
- Dies gilt beispielsweise für die Sende- und Empfangsleitung bei der V.24- oder RS-232-Schnittstelle, ebenso ist bei der IEC-625-Schnittstelle das gesamte Handshake als negative Logik ausgeführt.

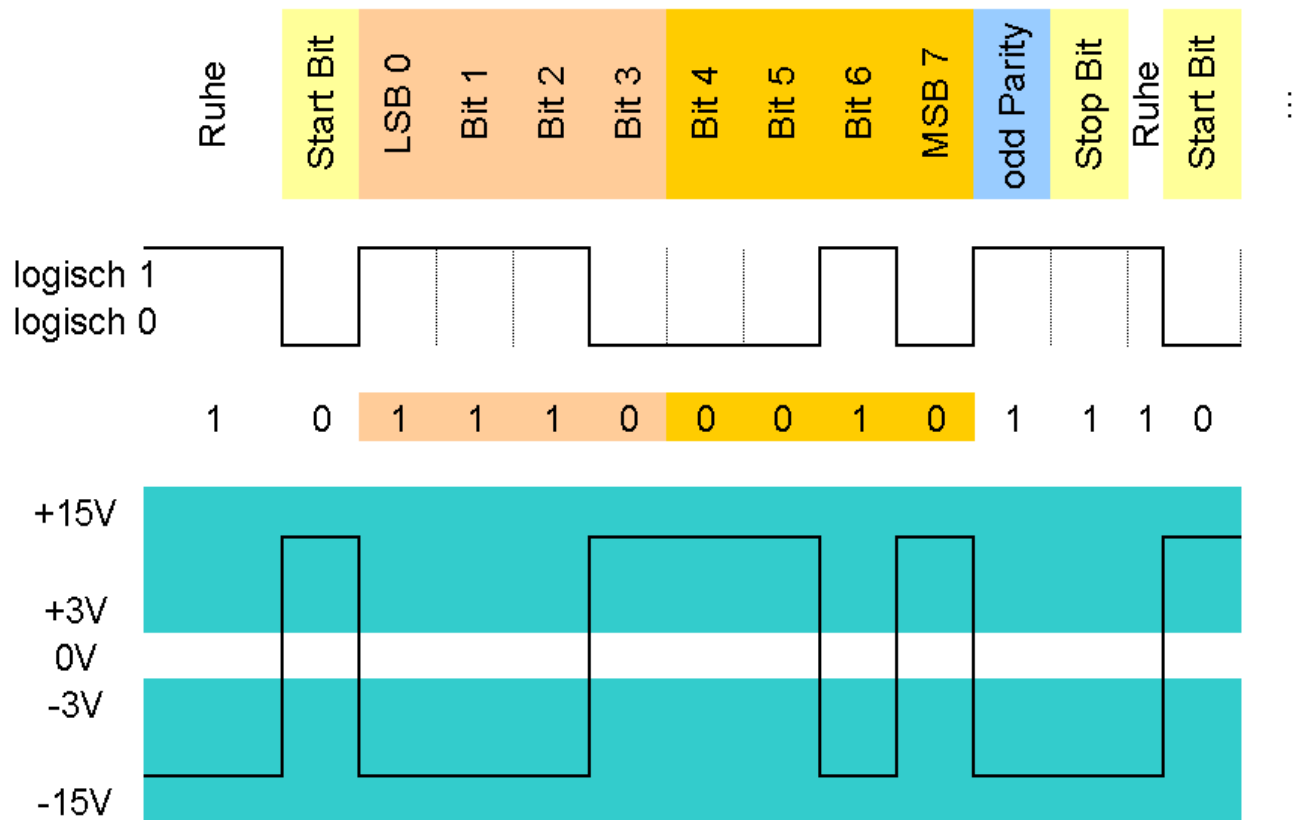
Synchronisation

Daten low & high

Check

9600 8O1 = 9600 Baud; 8 Datenbits; odd Parity; 1 Stopbit

ASCII "G" = \$47 = 0100 0111



Übertragungsparameter

- Zur Beschreibung der Übertragung einer RS232 Schnittstelle benötigt man folgende Parameter:
 - **Anzahl der Datenbits:** Üblicherweise 8 Bit. Teilweise aber auch 5,6,7 oder 9 Bit.
 - **Parität:** Nach den Datenbits kann ein Paritätsbit übertragen werden. Es kann gerade Parität (beschrieben durch E für *even*) oder ungerade Parität (O für *odd*) verwendet werden. Wird keine Parität verwendet, wird dies durch N beschrieben.
 - **Anzahl der Stoppbits:** Meist wird 1 Stoppbit verwendet. Es sind aber auch 2 Stoppbits möglich.
- Zusammen mit der Baudrate schaut eine Beschreibung zum Beispiel so aus:
38400 Baud 8O1 - sprich 38400 Baud mit 8 Datenbits, ungerade Parität und ein Stoppbit.

Universal Asynchronous Receiver Transmitter

- **Universal Asynchronous Receiver Transmitter**, kurz **UART**, ist eine elektronische Schaltung, die zur Realisierung digitaler serieller Schnittstellen dient.
- Dabei kann es sich sowohl um ein eigenständiges elektronisches Bauelement (ein UART-Chip bzw. -Baustein) oder um einen Funktionsblock eines höher-integrierten Bauteils (z. B. eines Mikrocontrollers) handeln.
- Eine UART-Schnittstelle dient zum Senden und Empfangen von Daten über eine Datenleitung und bildet den Standard der seriellen Schnittstellen an PCs und Mikrocontrollern.
- Auch im industriellen Bereich ist die Schnittstelle mit verschiedenen Interfaces (z. B. RS-232 oder EIA-485) sehr verbreitet.



STUDIENGÄNGE AM CAMPUS BAD MERGENTHEIM

Bad Mergentheim

I2C-Bus

I2C-Bus

- Beim I²C handelt es sich um einen Datenbus. Das bedeutet es handelt sich um ein Verfahren wie zwischen verschiedenen Teilnehmern kommuniziert wird.
- Beispielsweise wie ein μ C mit Displays und Sensoren kommuniziert.
- Das Besondere beim I2C ist das mit lediglich 2 Leitungen mit normalerweise bis zu 128 Teilnehmern kommuniziert werden kann
- Das Ganze funktioniert per serieller Datenübertragung, die Daten werden also hintereinander über eine Leitung geschickt und der I2C-Bus definiert den Ablauf wie das von staten geht

- Der Bus wurde 1982 von Philips eingeführt zur geräteinternen Kommunikation zwischen ICs in z. B. CD-Spielern und Fernsehgeräten.
- Dazu wurde die Mikrocontroller-Familie MAB8400 entwickelt, die einen I²C-Bus-Controller enthielt. Die erste standardisierte Spezifikation 1.0 wurde 1992 veröffentlicht.
- Diese ergänzte den ursprünglichen Standard mit 100 kbit/s um einen neuen „schnellen“ Modus (Fast-mode) mit 400 kbit/s und erweiterte den Adressraum um einen 10-Bit-Modus, so dass statt der ursprünglichen 112 Knoten seitdem bis zu 1136 unterstützt werden.

Übertragungsrate

0,1Mbit/s
(Standard)

Takt

Synchron

Datenübertragung

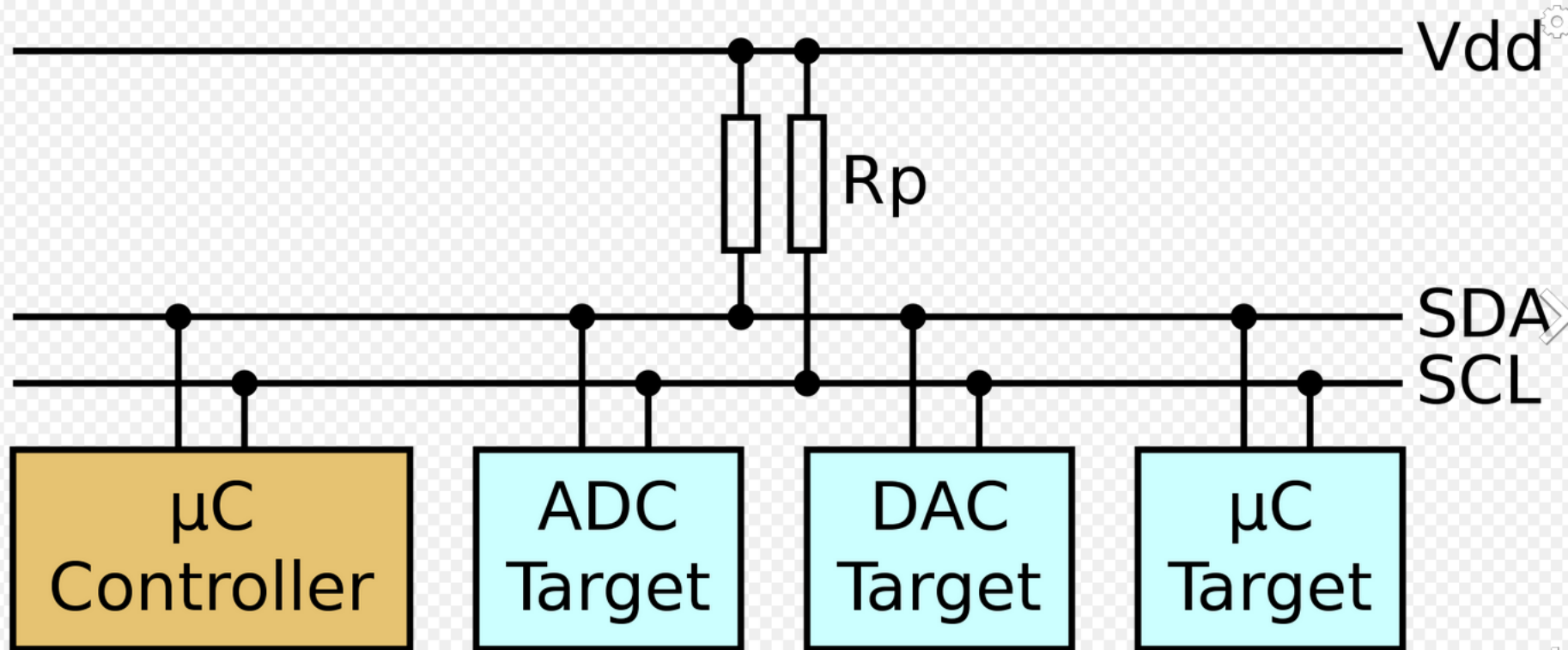
Seriell

Richtung

Bidirektional

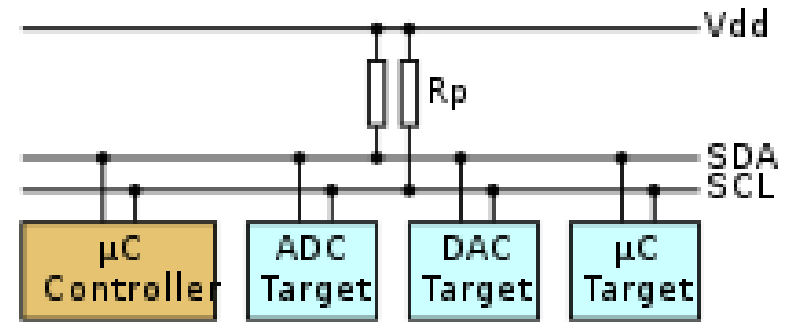
Bussystem

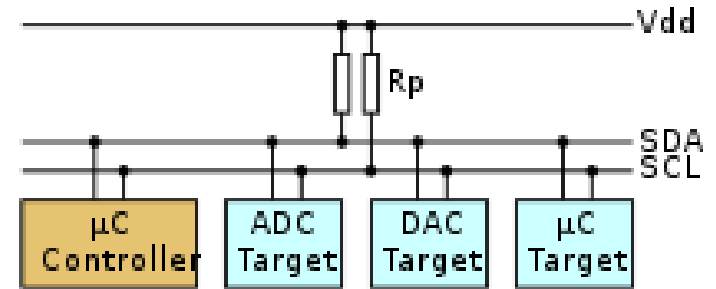
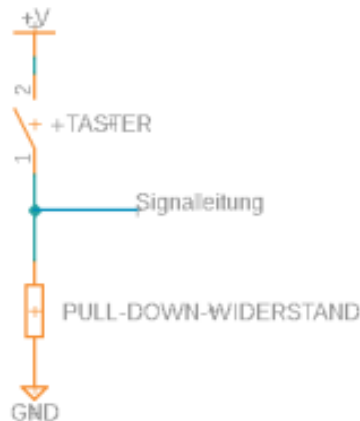
- I²C ist als Master-Slave-Bus konzipiert. Ein Datentransfer wird immer durch einen Master (Controller) initiiert; der über eine Adresse angesprochene Slave (Target) reagiert darauf.
- Ein Teilnehmer ist der *Master*, alle anderen sind die *Slaves*. Der Master hat als einziger das Recht, unaufgefordert auf die gemeinsame Ressource, den Bus, zuzugreifen.
- Der Slave kann von sich aus nicht auf die gemeinsame Ressource zugreifen; er muss warten, bis er vom Master gefragt wird (Polling) oder über eine an der gemeinsamen Ressource vorbei gehenden Verbindung dem Master anzeigen, dass er gefragt werden will.
- Mehrere Controller sind möglich (Multi-Controller-Betrieb). Wenn im Multi-Controller-Betrieb ein Controller-Baustein auch als Target arbeitet, kann ein anderer Controller direkt mit ihm kommunizieren, indem er ihn als Target anspricht.



Elektrische Definition

- Im Diagramm rechts sind drei Geräte eingezeichnet. I²C benötigt zwei Signalleitungen: Takt- (SCL = Serial Clock) und Datenleitung (SDA = Serial Data).
- Beide liegen mit den Pull-up-Widerständen R_p an der Versorgungsspannung V_{DD} .
- *Pull-up* bezeichnet in der Elektrotechnik einen (relativ hochohmigen) Widerstand, der eine Signalleitung mit dem höheren Spannungs-Potential verbindet.
- Durch ihn wird die Leitung auf das höhere Potential gebracht, für den Fall, dass kein Ausgang die Leitung *aktiv* auf ein niedrigeres Potential bringt.
- Übliche Werte liegen im Bereich von 1 k Ω bis rund 50 k Ω .

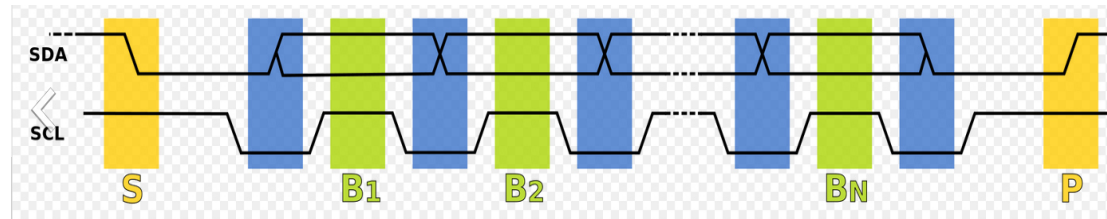




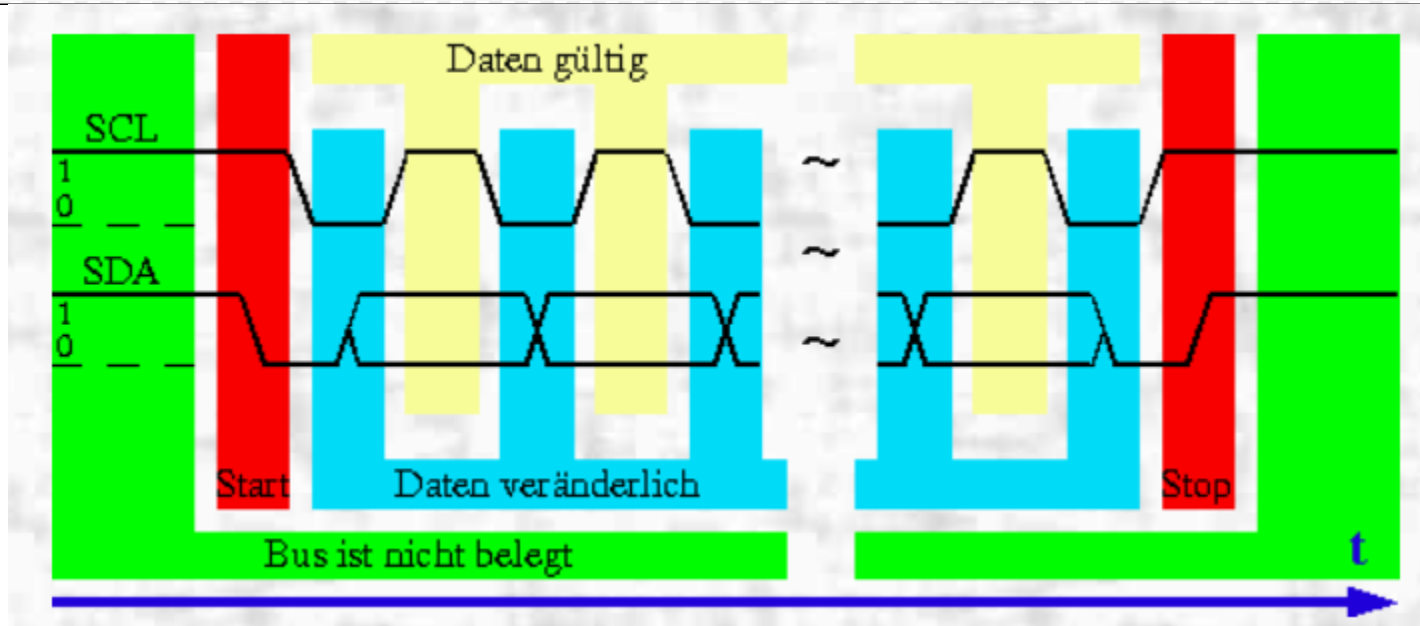
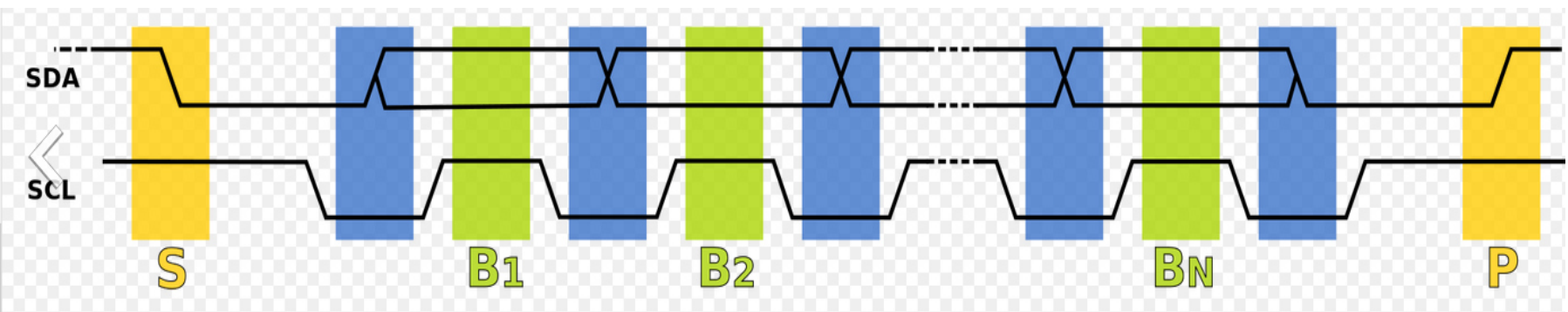
- Der High-Pegel soll mindestens $0,7 \times V_{DD}$ betragen, und der Low-Pegel soll bei höchstens $0,3 \times V_{DD}$ liegen.
- Die im Bild nicht eingezeichneten Serienwiderstände R_s an den Eingängen der Geräte sind optional und werden als Schutzwiderstände verwendet.
- Der I²C-Bus arbeitet mit positiver Logik, d. h. ein High-Pegel auf der Datenleitung entspricht einer logischen „1“, der Low-Pegel einer „0“.

Takt und Zustände des Busses

- Der Bustakt wird immer vom Controller ausgegeben.
- Das Taktsignal liegt nicht ständig an, sondern nur während der Datenübertragung.
- Für die verschiedenen Modi ist jeweils ein maximal erlaubter Bustakt vorgegeben.
- In der Regel können aber auch beliebig langsamere Taktraten verwendet werden, falls diese vom Controller-Interface unterstützt werden.
- Einige ICs (z. B. Analog-Digital-Umsetzer) benötigen jedoch eine bestimmte minimale Taktfrequenz, um ordnungsgemäß zu funktionieren.

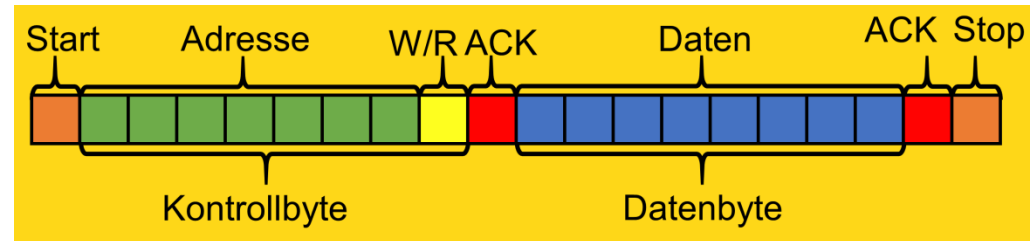


I2C-Bus

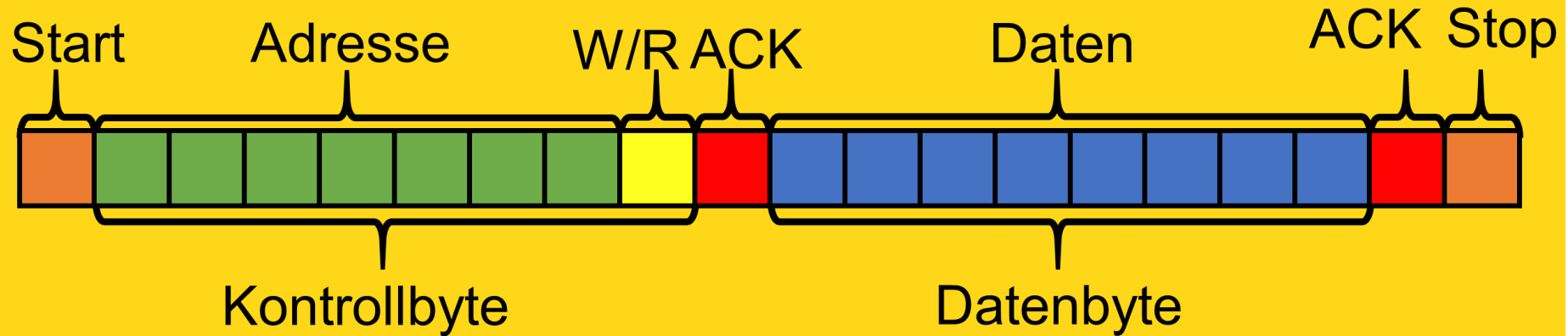


Ablauf

- Wie bereits erklärt werden bei I2C Bus Daten seriell gesendet.

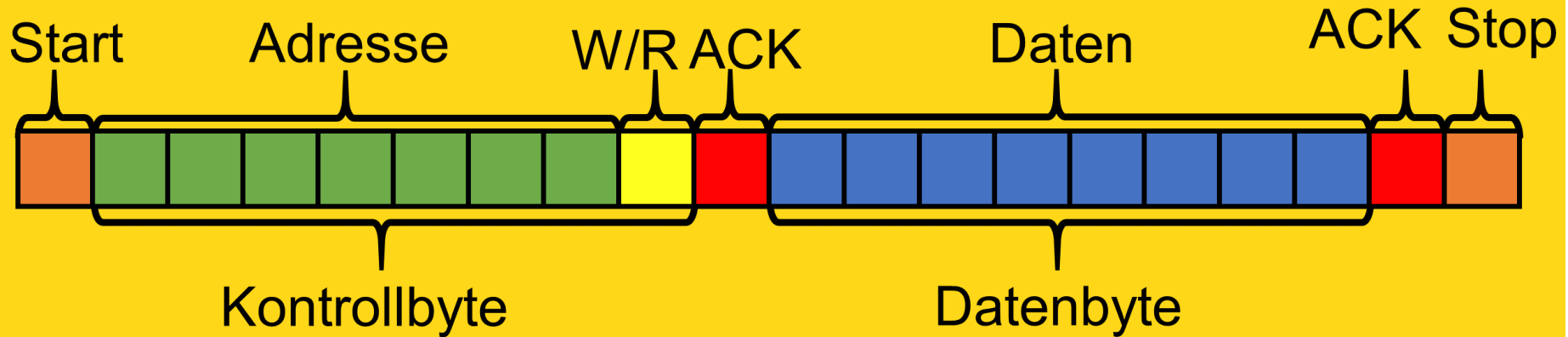


- Des bedeutet alle Daten werden hintereinander gesendet.
- Deshalb gibt es das sogenannte Protokoll welches genau vorschreibt in welcher Reihenfolge welche Daten geschickt werden.
- Das Protokoll ist also quasi so eine Art Anleitung oder Regelwerk damit alles geordnet von statten geht und jeder immer bescheid weiß wann was geschickt wird.



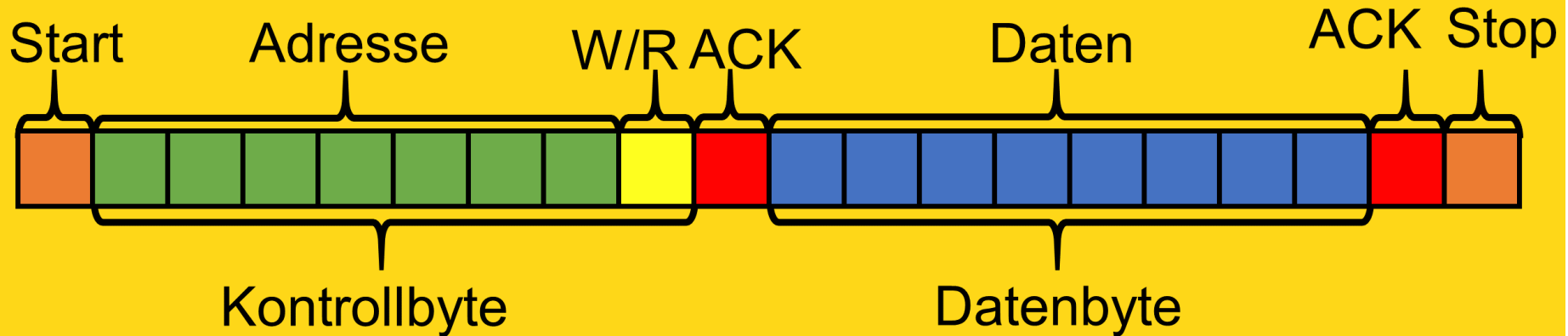
➤ Start-Condition

- Als erstes sendet der Master eine sogenannte „Start-Condition“.
- Dadurch werden alle Slaves hellhörig und hören genau zu was auf der Datenleitung folgend gesendet wird.



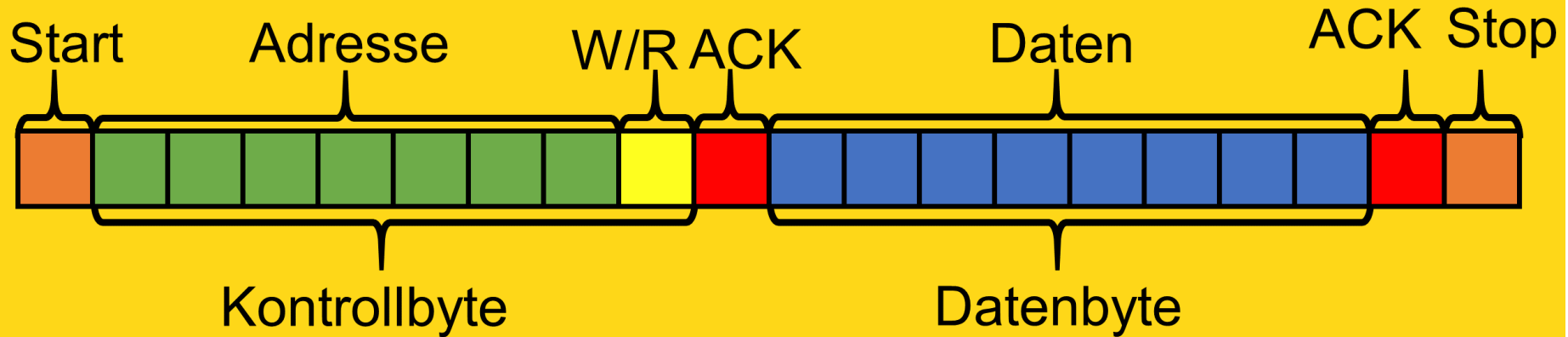
➤ Adresse

- Nach der Start-Condition sendet der Master nämlich die Adresse des Slaves mit dem er kommunizieren will.
- Diese Adresse besteht beim Standard I2C aus 7 Bit.
- Dann folgt die Adresse.
- Diese wird durch das ACK-Bit vom entsprechenden Target bestätigt.



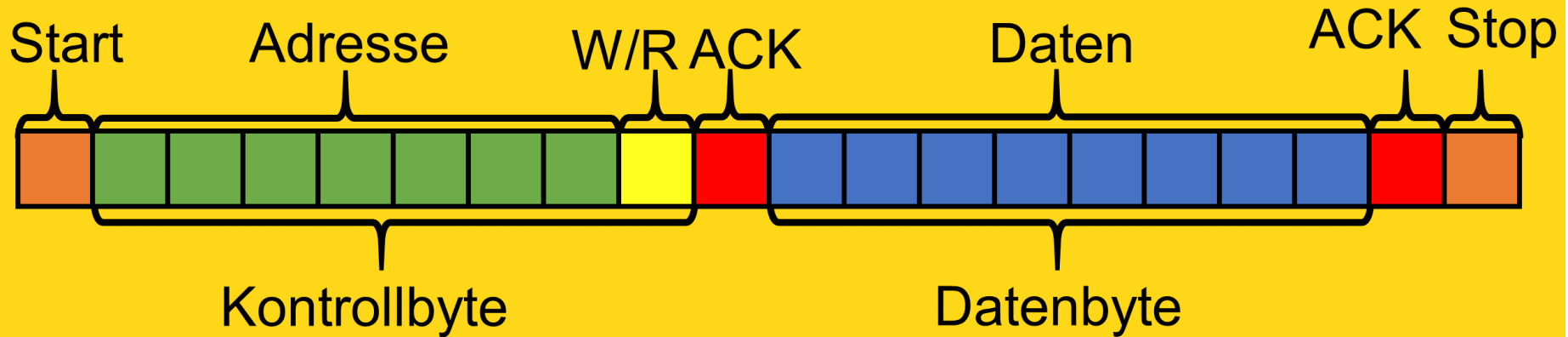
➤ Write / Read

- Nach der Adresse setzt er noch ein bit , write 0 oder read 1, spricht ob Daten vom Master zum Slave gesendet werden oder ob der Master Daten vom Slave haben möchte.
- Beispielsweise ob er die Temperatur eines Temperatur Sensors haben will oder ob er beispielsweise Daten zu einem Display sendet um etwas anzuzeigen.



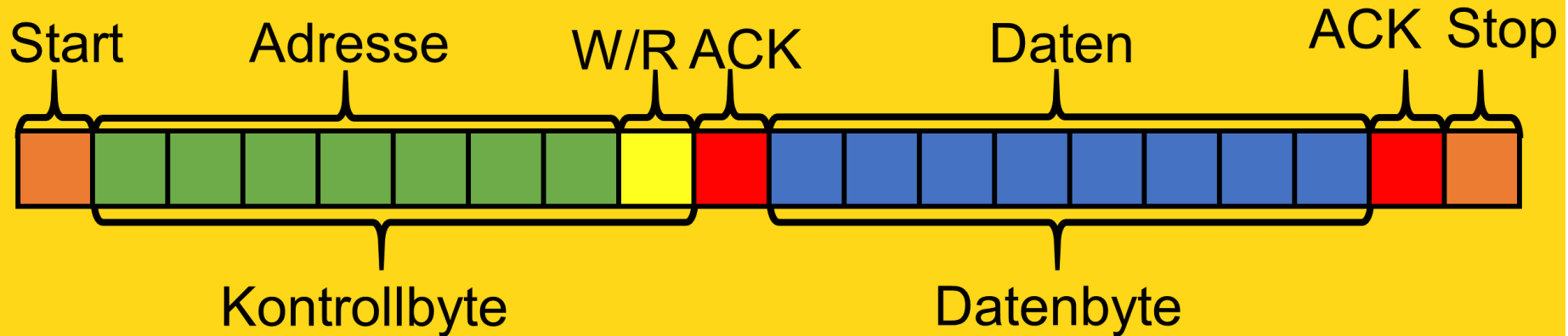
➤ Acknowledge

- Hat der Slave bis hier hin alles verstanden, sendet der Slave ein sogenanntes Acknowledge-Bit also eine Bestätigung das er bereit für die Kommunikation ist.
- 0 Bedeutet verstanden.
- Sobald der Master dieses Acknowledge empfängt sendet er Daten oder empfängt nach dem Acknowledge die Daten des Slaves, je nachdem wie das read/write bit gesetzt ist.



➤ Datenpaket

- Die Daten sind in sogenannten Datenpaketen verpackt.
- Jedes Datenpaket besitzt 8Bit also 1 Byte.
- Ist es sendendet folgt ein Acknowledge-Bit und darauf folgenden kommt das nächst Datenpaket.
- So können beliebig viele Datenpakete versende werden.



➤ Acknowledge

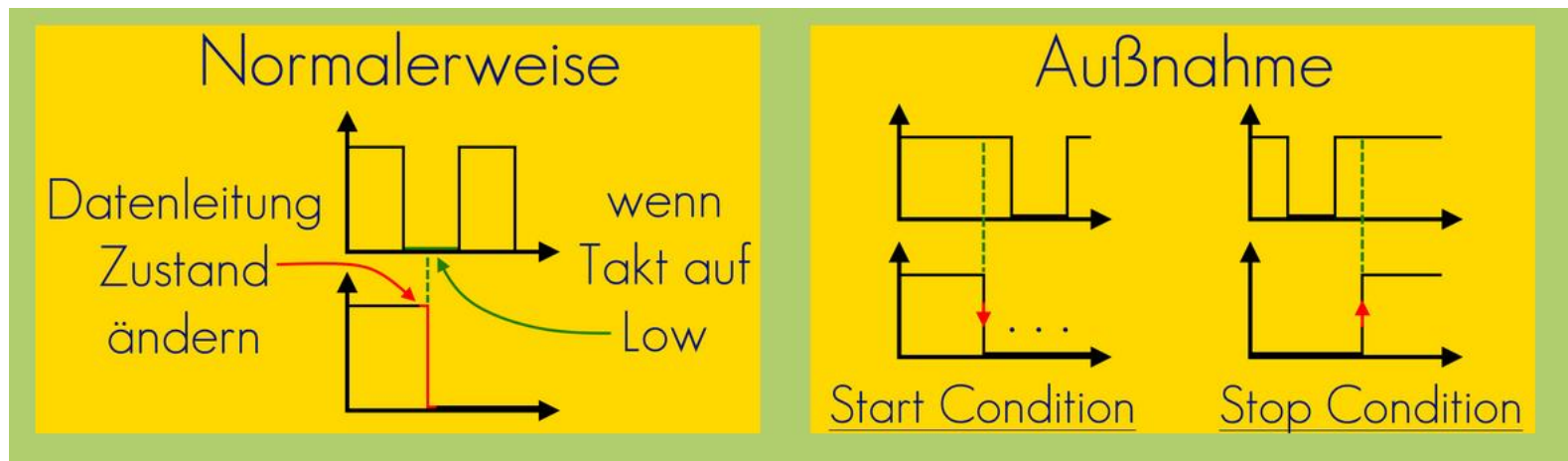
- Sobald alle Daten verschickt wurden, kommt das Acknowledge

➤ Stop-Condition

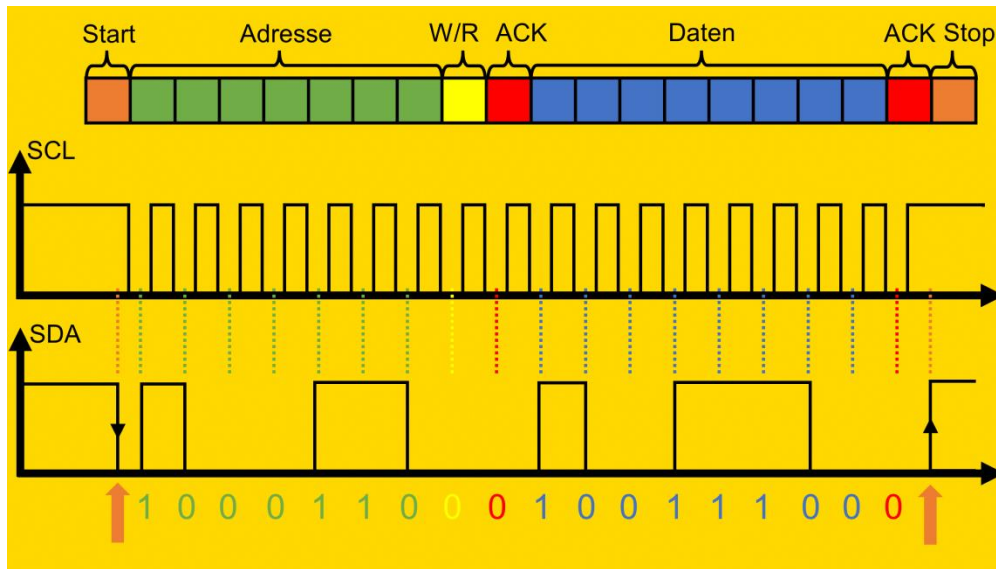
- Danach die sogenannte Stop-Condition.
- Dabei geht die Datenleitung auf High während wie Clockleitung auch auf High ist.

Start- Stop-Condition

- Normalerweise darf die Datenleitung nur den Zustand ändern, wenn der Takt auf Low ist, wie hier gezeigt.
- Eine Ausnahme wird bei der Start und Stop Condition gemacht.



Zeitdiagramm



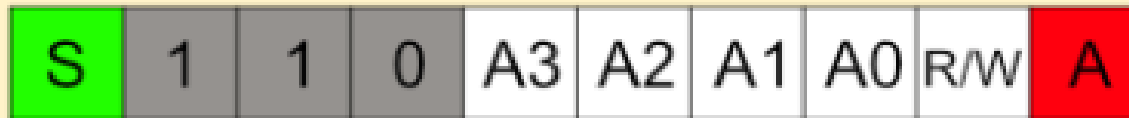
Start

- Bei Start wird der Dateneingang auf Low gesetzt, während die Taktleitung auf High ist.
- Dann ist klar die Kommunikation beginnt.
- Im Folgenden passiert alles wie bereits erklärt.

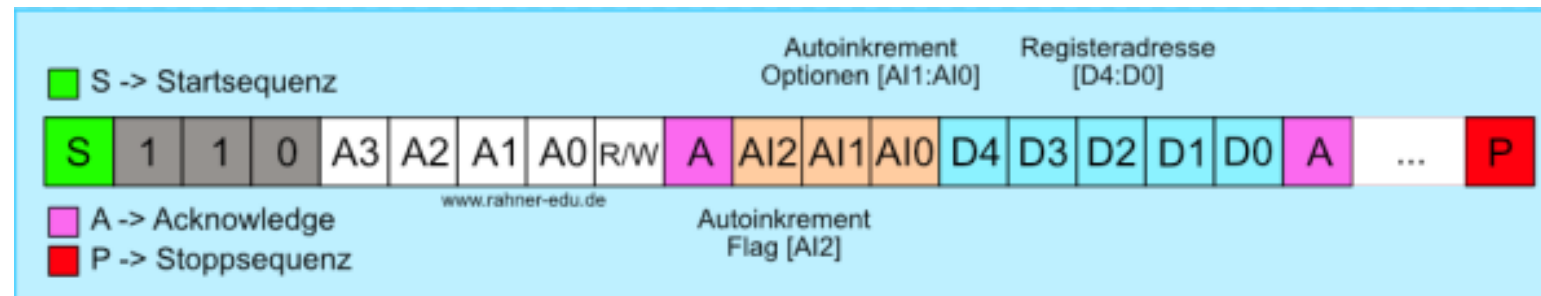
Adresse

- Es kommt die 7 Bit große Adresse, wie hier gezeigt, ganz klassisch 1 ist high und 0 ist Low.
- Dann kommt das Read/Write Bit, 0 für Write Schreiben und 1 für Read lesen, in diesem Beispiel wird also geschrieben also vom Master zum Slave gesendet.

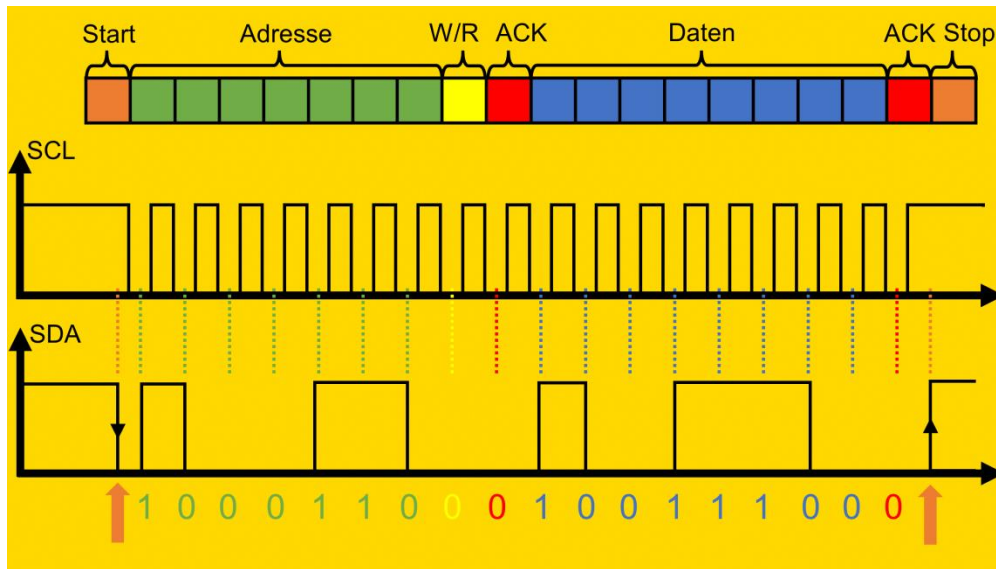
SLAVEADRESSE TLC59116



www.rahner-edu.de



Zeitdiagramm



Acknowledge-Bit

- Das nächste Bit ist das Acknowledge-Bit welches zur Bestätigung auf Low stehen muss.

Datenpaket

- Danach kommt das Datenpaket in Größe von einem Byte ebenfalls ganz klassisch 1 ist high und 0 ist Low.

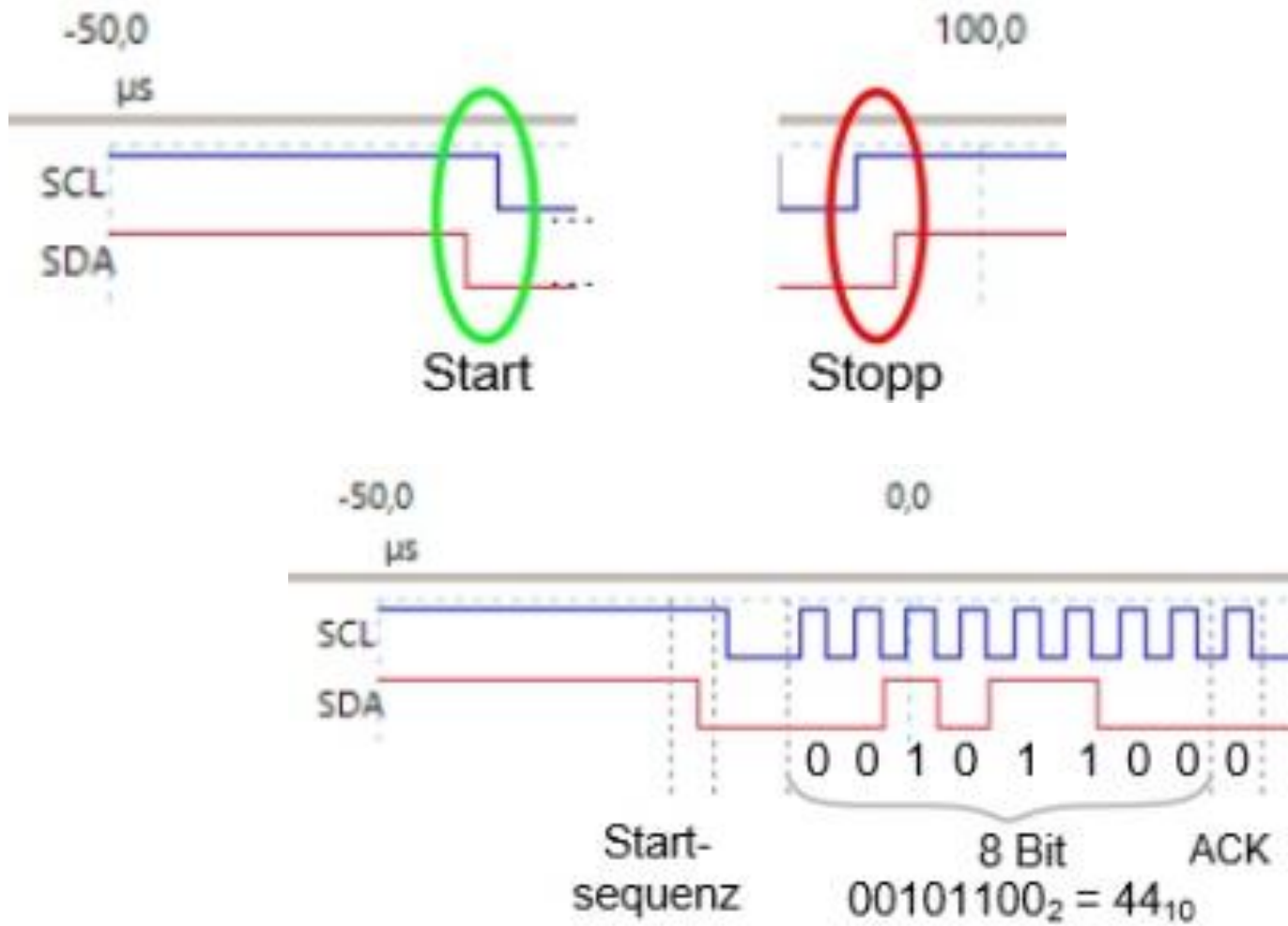
Acknowledge-Bit

- Nach dem Datenpaket folgt eine Bestätigung, das Acknowledge-Bit.

Stop

- Zum Schluss kommt die Stop-Condition hierzu geht die Datenleitung auf High während die Clockleitung auch auf High ist.

- Abhängig vom R/W-Bit werden nun Daten byteweise geschrieben (Daten an Target) oder gelesen (Daten vom Target).
- Das ACK beim Schreiben wird vom Target gesendet und beim Lesen vom Controller.
- Das letzte Byte eines Lesezugriffs wird vom Controller mit einem NACK quittiert, um das Ende der Übertragung anzuzeigen.
- Eine Übertragung wird durch das *Stop*-Signal beendet.
- Alternativ kann auch ein *Repeated-Start* am Beginn einer erneuten Übertragung gesendet werden, ohne die vorhergehende Übertragung mit einem *Stop*-Signal zu beenden.
- Alle Bytes werden dabei „Most Significant Bit First“ übertragen.



Verwendung

- Eine Eigenschaft von I²C ist die Tatsache, dass ein Mikrocontroller ein ganzes Netzwerk an integrierten Schaltungen mit nur zwei I/O-Pins und einfacher Software kontrollieren kann.
- Busse dieses Typs wurden realisiert, da ein nicht unerheblicher Teil der Kosten einer integrierten Schaltung und der verwendeten Leiterplatte von der Größe des Gehäuses und der Anzahl der Pins abhängt.
- Ein großes IC-Gehäuse hat mehr Pins, braucht mehr Platz auf der Leiterplatte und hat mehr Verbindungen, die versagen können. All das steigert die Produktions- und Testkosten.

Verwendung

- Obwohl langsamer als neuere Bus-Systeme, ist I²C wegen des geringen Aufwands vorteilhaft für Peripheriegeräte, die nicht schnell zu sein brauchen.
- Häufig wird er für die Übertragung von Steuer- und Konfigurationsdaten verwendet.
- Beispiele sind Lautstärkeregler, Analog-Digital- oder Digital-Analog-Wandler mit niedriger Abtastrate, Echtzeituhren, kleine, nichtflüchtige Speicher oder bidirektionale Schalter und Multiplexer.
- Auch elektronische Sensoren haben oft einen Analog-Digital-Wandler mit I²C-Schnittstelle integriert.
- Während des Betriebes können Chips zum Bus hinzugefügt oder entfernt werden (Hot-Plugging).

Serielles EEPROM mit I²C-Bus von STMicroelectronics





E N D E