

Diese Webseite verwendet Cookies. Cookies helfen dabei, diese Webseite besser zu machen. [mehr Informationen](#) [Ok](#)
I2C einverstanden

Aus RN-Wissen.de



IIC, Inter-IC bzw. Inter Integrated Circuit Bus

Inhaltsverzeichnis

- 1 Geschichte
- 2 Bitübertragung
- 3 Start- und Stoppbedingungen
 - 3.1 Startbedingung
 - 3.2 Stoppbedingung
 - 3.3 Repeated-Startbedingung
- 4 Datenübertragung größerer Datenmengen
 - 4.1 Byteübertragung
 - 4.2 Bestätigung (Acknowledgment)
- 5 Adressierung
 - 5.1 7-Bit Adressierung
 - 5.1.1 Aufbau einer 7-Bit-Adresse
 - 5.1.2 Reservierte Adressen
 - 5.1.3 Subadressen
 - 5.2 10-Bit-Adressierung
- 6 Beispiel einer Datenübertragung
 - 6.1 Schema
 - 6.2 Oszillogramm
 - 6.3 Programmierung Beispiel
- 7 Siehe auch
- 8 Weblinks

Geschichte

Der I2C-Bus (*Inter Integrated Circuit*) ist ein serieller synchroner Zweidraht-Bus, der vor rund 20 Jahren von Philips entwickelt wurde. Was die Philips-Designer seinerzeit kaum ahnen konnten: Ihr Inter-IC Bus (I2C, "1 Squared C") ist mittlerweile ein Industriestandard für Steuerungs-, Diagnose- und Überwachungslösungen in unzähligen Embedded-Applikationen. Mit einfacher Implementierung, niedrigen Kosten und einer Übertragungsrate bis 3,4 MBit/s ist der Bus-Veteran aktuell wie nie zuvor.

Ziel der damaligen Entwicklung war ein hierarchisches Bus-System, über das mehrere ICs bei geringstem Aufwand (Leiterbahnen, Komponenten, etc.) miteinander kommunizieren sollten. Daher bot sich eine serielle Struktur an, um

- im Vergleich zu parallelen Bus-Systemen
- mit weniger Leiterbahnen und I/Os auszukommen und so letztendlich Platinenfläche, IC-Abmessungen (Pin-Anzahl) und Kosten zu reduzieren.

Aus diesen Überlegungen entstand schließlich ein echter bidirektionaler Zweidraht-Bus in Master/Slave-Architektur mit integriertem Übertragungsprotokoll und Software-Adressierung, der nur zwei Verbindungen zwischen den ICs/Boards erfordert: Die Taktleitung SCL (Serial Clock Line) und die Datenleitung SDA (Serial Data Line).

Das bedeutet in der Praxis, dass z.B. ein Microcontroller ein ganzes "Netzwerk" von Chips mit nur zwei I/O Pins und einfacher Software steuern bzw. für bestimmte Funktionen nutzen kann. Ursprünglich wurde der I2C-Bus für Interaktionen zwischen einigen wenigen ICs entwickelt, die auf derselben Platine montiert waren, etwa zur Steuerung der Abstimmung von Autoradios oder TV-Geräten.

Dabei betrug die Übertragungsrate nur 100 kBit/s bei einer zulässigen Bus-Kapazität von 400 pF. Den ständig steigenden Leistungsanforderungen folgend, wurde die Übertragungsrate 1992 und 1998 auf 400 kBit/s bzw. 3,4 Mbit/s angehoben. Ein weiterer großer Vorteil des I2C-Bus besteht darin, dass auch deutlich langsamere Busteilnehmer am Bus betrieben werden können. Siehe dazu Clock Stretching.

Heute wird der I2C-Bus auch in Systemen mit mehreren Boards wie Blade- oder Rack-Mount-Servern eingesetzt, wobei I2C-Hot-Swap-Buffer ein störungsfreies Einsetzen oder Austauschen von Boards während des Betriebs gewährleisten. Zum Beispiel wird der I2C-Bus heute auch bei nahezu allen Projekten aus dem Roboternetz verwendet. Siehe dazu Projekte und Schaltungen.

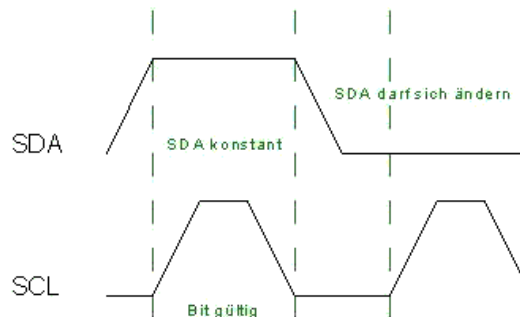
Mit Hilfe neuer Expansions- und Steuerungs-Bauelemente kann der I2C-Bus inzwischen über die 400-pF-Grenze (ca. 20 bis 30 ICs pro Bus-Segment) hinaus erweitert werden. Dadurch können Entwickler mehr Chips **sogar mehrere identische ICs mit derselben Adresse** anschließen und flexibel auf die steigende Zahl von I2C-Bauelementen reagieren.

Ein großer Vorteil des I2C-Bus ist auch die einfache Ansteuerung. Da keine festen Taktzeiten eingehalten werden müssen, können sowohl langsame als auch sehr schnelle Busteilnehmer, Chips und Programmiersprachen eingesetzt werden.

<http://www.roboternetz.de/bilder/i2cbeitrag1.gif>

Bitübertragung

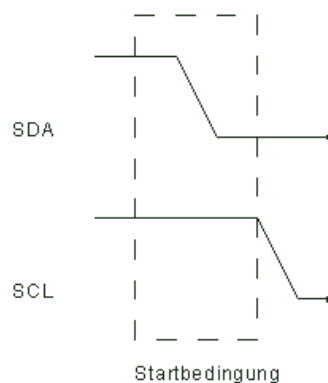
Um ein Bit als gültig zu werten, muss SCL High sein. SDA darf sich währenddessen nicht ändern (es sei denn es handelt sich um die Start- oder Stoppbedingung, doch dazu später mehr). Um beispielsweise eine 1 zu übertragen, müssen SDA sowie SCL High sein. Für eine 0, muss SDA Low sein, SCL jedoch High.



Start- und Stoppbedingungen

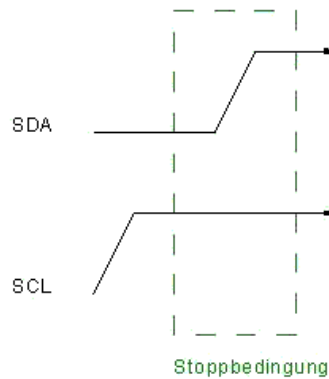
Startbedingung

Um die angeschlossenen ICs zu informieren, dass eine Datenübertragung beginnt, muss eine Startbedingung erzeugt werden. Vorher kann keine Datenübertragung erfolgen. Eine Startbedingung wird erzeugt, indem, während SCL High ist, SDA von High auf Low wechselt.



Stoppbedingung

Die Stoppbedingung funktioniert genau anders herum: SCL muss High sein und während dieser Phase wechselt SDA von Low auf High. Die Stoppbedingung beendet, wie der Name schon vermuten lässt, eine Datenübertragung. So kann der Master signalisieren, dass er keine weiteren Daten empfangen oder senden möchte.



Repeated-Startbedingung

Da die Stoppbedingung gleichzeitig auch eine Freigabe des Bus bedeutet (und dann könnte ja ein anderer Master den Bus übernehmen), gibt es auch den Start ohne vorheriges Stopp. Das wird dann benötigt, wenn vor dem Lesen erst ein Argument/Command geschickt werden muß. Abfolge:

```
I2C Start
I2C Send Write-Address
I2C Send Argument
I2C Start oder Repeated Start
I2C Send Read-Address
I2C Read Data
....
I2C Stopp oder Release Bus
```

Datenübertragung größerer Datenmengen

Byteübertragung

Wenn ein Byte verschickt werden soll, dann wird als erstes das höchstwertigste Bit verschickt. Dann folgen die anderen bis hin zum niederwertigstem. Folgende Tabelle soll dieses Schema anhand der Zahl 109 verdeutlichen:

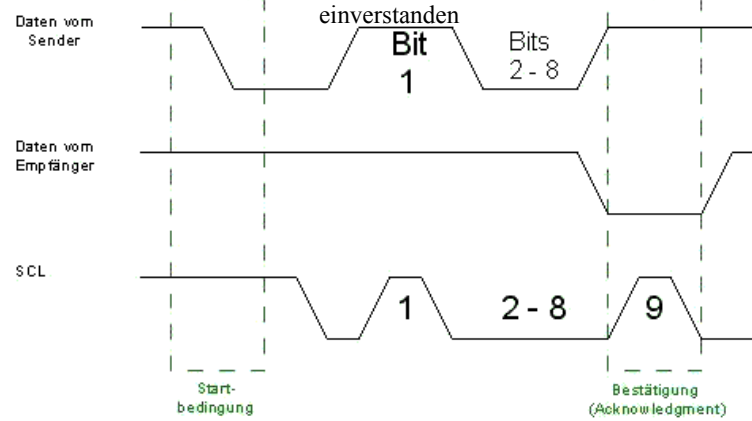
Dezimal	128	64	32	16	8	4	2	1
Dual	0	1	1	0	1	1	0	1
Übertragungsreihenfolge	1	2	3	4	5	6	7	8

Wie man sieht, wird das Bit, das die Dezimalzahl 128 repräsentiert, als erstes übertragen.

Bestätigung (Acknowledgment)

Der Empfänger quittiert den Erhalt der Daten mit einer Bestätigung (oder auch Acknowledgment). Nach acht Datenbits und folglich auch acht Taktimpulsen wird eine Bestätigung erzeugt.

Diese Webseite verwendet Cookies. Cookies helfen dabei, diese Webseite besser zu machen. mehr Informationen Ok



Der Empfänger "zieht" SDA auf Low, bis der Master den neunten Taktimpuls generiert hat. Findet diese Bestätigung statt, bedeutet das auch gleichzeitig, dass der Empfänger ein weiteres Byte empfangen möchte. Möchte der Empfänger kein weiteres Byte mehr empfangen, dann verschickt er keine Bestätigung. Das eigentliche Ende der Datenübertragung wird aber durch die Stoppbedingung ausgelöst (Oder die "Repeated Startbedingung").

Adressierung

Das erste Byte nach der Startbedingung, das der Master verschickt, ist die Adresse des Slaves, den er ansprechen möchte.

7-Bit Adressierung

Die 7-Bit Adressierung ist die erste Adressierungsform des I²C Busses und ermöglicht prinzipiell bis zu 128 (2^7) Geräte an einem Bus. Durch die Reservierung einiger Adressen, unter anderem für die 10-Bit-Adressierung, ist die 7-Bit-Adressierung auf 112 Geräte begrenzt und ermöglicht so eine konfliktfreie Adressierung des 7-Bit- und des 10-Bit-Adressraumes (siehe Tabelle unten: Reservierte Adressen).

Aufbau einer 7-Bit-Adresse



Auch hier wird mit dem wichtigstem Bit begonnen. Da ein Byte aber acht Bits hat, und die Adresse nur sieben Bits, gibt es noch ein Bit, das die Datenrichtung angibt. Es bestimmt, ob der Master Daten empfangen möchte oder ob er dem Slave Daten schicken möchte. Dieses letzte Bit wird als R/W Bit bezeichnet (0 steht für "schreiben", 1 für "lesen"). Hat ein Slave seine Adresse richtig verstanden und ist bereit Daten zu empfangen oder zu verschicken, dann sendet er eine Bestätigung. Hat er seine Adresse nicht richtig mitbekommen, oder ist gerade nicht in der Lage, Daten zu verschicken oder zu empfangen, dann bleibt die Bestätigung aus. Der Master kann dann eine Stoppbedingung erzeugen, so dass die Datenübertragung abgebrochen wird.

Reservierte Adressen

Zwei Adressbereiche von jeweils 8 Adressen (1111XXX und 0000XXX) wurden reserviert, um den Bus ausbaufähig zu halten oder um Missverständnissen vorzubeugen.

Adresse	R/W Bit	Beschreibung
0000000	0	General Call Adresse
0000000	1	Startbyte
0000001	X	CBUS Adresse
0000010	X	Reserviert für ein anderes Busformat
0000011	X	Für zukünftige Erweiterungen reserviert

Diese Webseite verwendet Cookies. Cookies helfen dabei, diese Webseite besser zu machen. [mehr Informationen](#) [Ok](#)

00001XX	X	Für zukünftige Erweiterungen reserviert
11111XX	X	Für zukünftige Erweiterungen reserviert
11110XX	X	10-Bit Adressierung

Nach Abzug der 16 Reservierten Adressen bleiben also 112 freie Adressen übrig, die man mit der 7-Bit-Adressierung verwenden kann.

Subadressen

Als Subadressen bezeichnet man die vom Benutzer programmierbaren Adressbits einer Adresse. Das sind bei den meisten ICs die letzten drei Adressbits. Da man so acht verschiedene Adresskombinationen bekommt (2^3), kann man folglich maximal acht dieser ICs an einem Bus gleichzeitig nutzen. Folgende Tabelle verdeutlicht das am Beispiel des I²C-Bausteins PCF 8570:

1. Teil: Feste Adresse	2. Teil: Subadresse
1010	XXX

10-Bit-Adressierung

Da es mit der Zeit immer mehr I²C-Bausteine gab und es schon vorher Überschneidungen bei den Adressen gab, wurden 10-Bit-Adressen eingeführt. Diese erlauben bis zu 1024 (2^{10}) Geräte an einem Bus. Durch die Reservierung der Adressen 1111 0XX und des R/W Bits dieser Adresse, werden mögliche 7-Bit-Geräte am Bus nicht gestört. Im Gegenteil, es können sogar 7-Bit- und 10-Bit-Geräte an einem Bus betrieben werden.

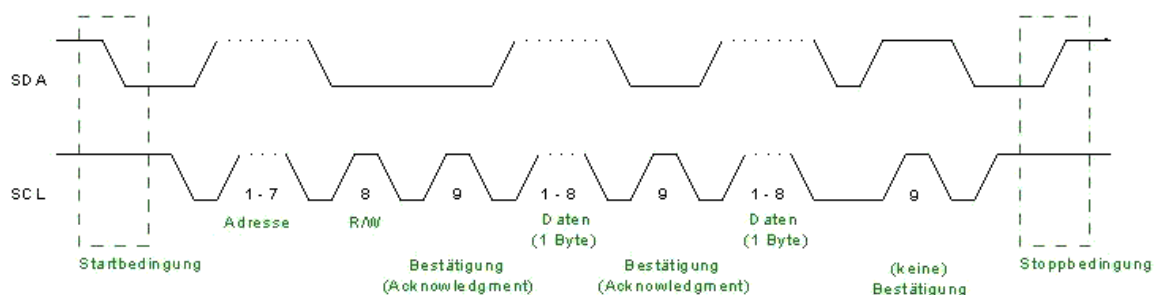
Reservierte Adresse	Adresse Teil 1	R/W Bit		Adresse Teil 2	
11110	XX	X	1. Bestätigung	XXXXXXXX	2. Bestätigung

Als erstes wird die reservierte Adresse mit den ersten fünf Bits "11110" gesendet. Dadurch wird – wie schon gesagt – das Stören von Bausteinen mit 7-Bit-Adressen vermieden. Nun folgen die ersten zwei Bits der eigentlichen Adresse. Dann kommt schon das R/W Bit, da nach einem Byte immer eine Bestätigung folgen muss, wenn eine weitere Datenübertragung erwünscht ist. Da es vorkommen kann, dass mehrere Teilnehmer auf den ersten Teil der Adresse reagieren, werden auch alle diese betroffenen Teilnehmer die Bestätigung generieren. Nach der ersten Bestätigung kommt der 2. Teil der Adresse, der diesmal ein ganzes Byte ist. Alle Teilnehmer, bei denen der erste Adressteil schon zugetroffen hat, werden auch das zweite Adressbyte überprüfen, aber nur der nun adressierter Teilnehmer wird eine Bestätigung erzeugen. Nach der zweiten Bestätigung kann der eigentliche Datenaustausch beginnen.

Beispiel einer Datenübertragung

Schema

Zum Abschluss noch das schematische Beispiel einer vollständigen Kommunikation zwischen einem Mikrocontroller als Master und einem Slave mit 7-Bit-Adresse. In diesem Fall wird schreibend auf den Slave zugegriffen. Der Master ist also ein Sender, während der Slave als Empfänger fungiert.



Nach der Startbedingung folgt die Slaveadresse mit dem R/W Bit. Dann kommt die obligatorische Bestätigung und der Master sendet ein Byte. Nun folgt wieder eine Bestätigung und der Master sendet wiederum ein Byte. Der Slave sendet wieder seine Bestätigung, jedoch möchte der Master jetzt keine Daten mehr senden, da SDA jetzt auf +5 Volt liegt. Da nun das Ende der Datenübertragung erreicht ist, löst der Master die Stopbedingung aus.

Oszillogramm

Hier ein typisches Bild einer I2C-Kommunikation: Adresse gesendet aber kein ACK.

http://www.rn-wissen.de/images/9/92/I2c_oszi-IMAG0186.jpeg

Programmierung Beispiel

Ein typisches Beispiel wie in einer Programmiersprache (hier Bascom Basic) Daten über den I2C-Bus verschickt werden:

```
i2c_init
i2c_start
i2c_sendbyte(slaveid)
i2c_sendbyte(wert1)
i2c_sendbyte(wert2)
i2c_stop
```

Autoren:

Frank
Vish

Verwendete Quellen für diesen Text:

1. Philips Infos
2. Spoerle Produktinfos
3. mehrere Roboternetz-Artikel

Siehe auch

- RN-Definitionen
- Clock Stretching
- I2C Chip-Übersicht
- LCD an I2C Huckepack auf RN-Miniplatine
- Two-wire Serial Interface (TWI) bei AVR Megas
- TWI-Slave mit avr-gcc

Weblinks

- Spezifikationen (http://www.roboternetz.de/phpBB2/dload.php?action=file&file_id=28)
- PC -> I2C-Bus-Adapter (<http://www.roboternetz.de/phpBB2/viewtopic.php?t=4509>)
- I2C über 50 Meter mit I2C-Bus-Extender möglich (IC P82B715) (http://www.nxp.com/products/interface_and_connectivity/i2c/i2c_bus_repeater_hubs_extenders/series/P82B715.html)
- i2c-bus.org - Hilfreiche I2C-Informationen / Englisch (<http://www.i2c-bus.org/>)
- I2C Technical Overview and Frequently Asked Questions / Englisch (<http://www.esacademy.com/faq/i2c/index.htm>)
- 3,3V und 5V-Bausteine am selben I2C-Bus (<http://www.elektronikwissen.net/pegelwandler/7-bidirektionaler-levelshifter-fuer-i2c.html>)


Von „<http://rn-wissen.de/wiki/index.php?title=I2C&oldid=24645>“

Diese Webseite verwendet Cookies. Cookies helfen dabei, diese Webseite besser zu machen. [mehr Informationen](#) [Ok](#)

The PCF8574 from TI

8-Bit Input/Output Expander for The Two-Line Bidirectional Bus.



[Datenschutzinfo](#) 

[► I2C Protocol](#)

[► I2C Bus](#)

[► I2C Interface](#)

[► I2C SPI](#)

Kategorien: [Kommunikation](#) | [Microcontroller](#) | [Robotikeinstieg](#) | [Grundlagen](#) | [Elektronik](#) | [Abkürzung](#)