

# **I2C-BUS**



**Von Ramesh Sathiyamoorthy  
Klasse E4p  
Embedded Control  
Hr.Felser  
HTI Burgdorf**

# Inhaltverzeichnis

Inhaltverzeichnis .....	2
1.History .....	3
2.Einsatzbereich .....	4
3.Funktionweise .....	5
3.1 Das I2C-Busprotokoll .....	5
3.2 Die Datenübertragung zwischen Master und Slave .....	7
3.2.1 Datenübertragung vom Master zum Slave .....	7
3.2.2 Datenübertragung vom Slave zum Master .....	8
3.2.3 Kombinierte Datenübertragung zwischen Master und Slave .....	9
4.Installationstechnik.....	10

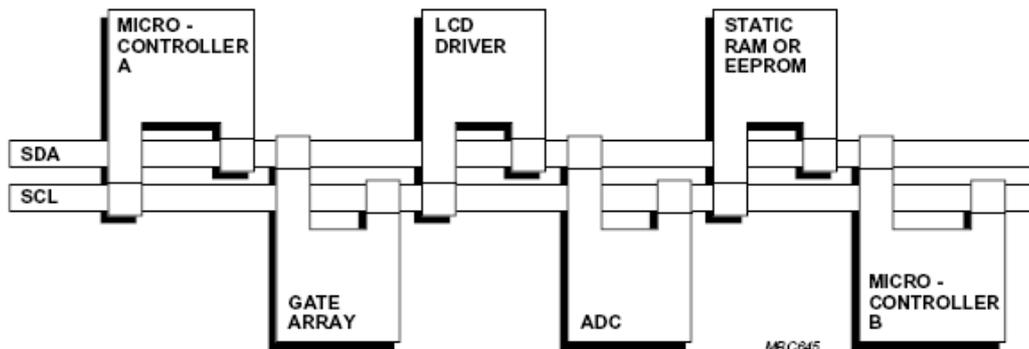
# 1. History

Der I<sup>2</sup>C-Bus wurde 1982 von Philips semiconductors spezifiziert, um in Geräten der Unterhaltungselektronik auf eine einfache Art Steuerungsaufgaben durchzuführen. I<sup>2</sup>C-Bus steht akronym für **Inter-Integrated-Circuit-Bus. Diese Bezeichnung erklärt die Absicht der Entwicklung: Der I<sup>2</sup>C-Bus dient der Kommunikation zwischen integrierten Schaltkreisen. Seit der Einführung des I<sup>2</sup>C-Busses durch Philips sind zahlreiche Integrierte Bausteine mit I<sup>2</sup>C-Spezifikation entwickelt worden. Viele andere Halbleiterhersteller (Siemens, Texas Instruments, SGS-Thomson, Maxim usw.) fertigen ebenfalls Chips, die den I<sup>2</sup>C-Bus nutzen. Einige Chip-Typen sind auf die Bedürfnisse der Unterhaltungselektronik zugeschnitten. Wegen der weiten Verbreitung und der Störunanfälligkeit des Busses wurden auch Schaltkreise für die allgemeine Elektronik, die Mess- und Datentechnik entwickelt, z.B.:**

- Analog-Digital-Wandler
- Digital-Analog-Wandler
- EEPROM-Speicher-Bausteine
- Echtzeit-Uhren und Timer
- Display-Treiber
- Frequenz-Synthesizer
- Parallele I/O-Ports

## 2.Einsatzbereich

I<sup>2</sup>C ist ein synchroner serieller Zweidraht-Bus (eine Daten- und eine Taktleitung), der für die Kommunikation zwischen ICs über kleine Distanzen geeignet ist. Gesprochen "I quadrat C" steht für IIC = Inter IC Bus. Aus Lizenzgünden heißt der I<sup>2</sup>C Bus bei manchen Herstellern auch TWI, two wire interface.



In einem I<sup>2</sup>C-Bus gibt es mindestens einen Master und eine beliebige Anzahl Slaves (max. 128). Ein I<sup>2</sup>C-Bus mit mehreren Mastern wird als "Multi-Master-Bus" bezeichnet. Der (oder die) Master sprechen die Slaves an; ein Slave kann NIE selbstständig Daten senden. Dazu 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 Protokoll sicher, dass sich diese nicht gegenseitig stören.

Die Übertragungsrate beträgt beim Standard mode bis zu 100 kbit/s, beim Fast mode bis zu 400 kbit/s und beim High-speed mode bis zu 3,4 MBit/s. Falls die Taktrate für einen Slave zu hoch ist, kann er die Clock-Leitung auf Null ziehen und die Übertragung damit verlangsamen. Dies ist auf Bit- wie auf Byte-Ebene möglich, Ersteres allerdings nicht im High-speed mode. Um längere Übertragungswege zu realisieren, kann man die Taktrate fast beliebig vermindern (einige Bausteine erzeugen aber irgendwann ein Time-Out). Mit einer Taktfrequenz von nur 5 kbit/s können dann durchaus mehrere Meter überbrückt werden.

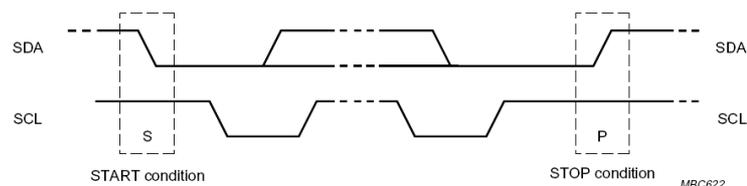
Physikalisch handelt es sich um ein bidirektionalen seriellen Zweidraht-Bus, bei der eine der beiden Leitungen (Shift DATA - Leitung: SDA) zum Datentransport,

die andere (Shift Clock-Leitung: SCL) zur Synchronisation des Datenverkehrs dient. Genau genommen benötigt man noch eine dritte Leitung, die Masseleitung (GND), die als Bezugspotential mitgeführt werden muss. An den Bus angeschlossen sind mindestens ein Master-Baustein, der den Datenverkehr auf dem Bus regelt und eine beliebige Anzahl Slave-Bausteine, die Daten vom Bus empfangen oder auf den Bus legen. Grundsätzlich ist der Master für die Generierung der Shift-Clock-Signale zuständig, mit den die Daten auf der Shift-Data-Leitung weitergetaktet werden. Die Slaves können entweder Empfänger (Receiver) von Daten, oder Sender (Transmitter) von Daten oder beides (Transceiver) sein. Ein typisches Receiver-Baustein, also ein Baustein, der nur Daten empfängt, wäre etwa ein Anzeigentreiber. Ein EEPROM-Baustein wäre ein Beispiel für einen Transceiver, den es sollen ja sowohl Daten in den Speicherbaustein eingelesen als auch wieder ausgelesen werden.

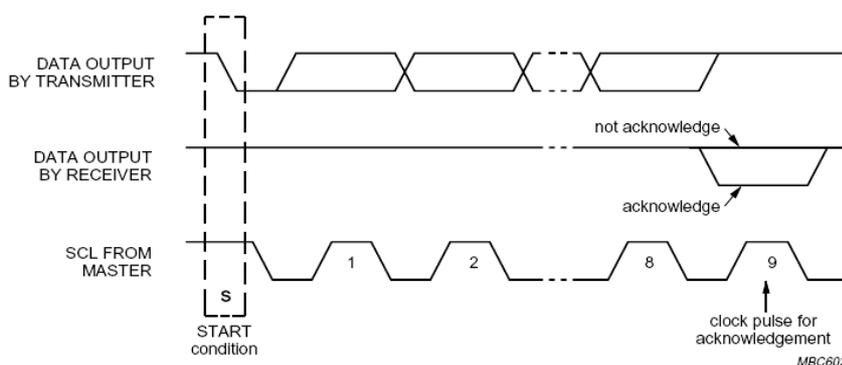
### 3.Funktionweise

#### 3.1 Das I<sup>2</sup>C-Busprotokoll

Im I<sup>2</sup>C-Bus erfolgt die Datenübertragung bitseriell und synchron, d.h. daß jedes Datenbit auf der SDA-Leitung mit dem Takt der SCL-Leitung synchronisiert wird. Vor Beginn der Kommunikation haben beide Busleitungen SCL und SDA zunächst High-Pegel d.h. alle Bausteine am Bus sind im "Schlaf-Modus". Nun wird die CPU aktiv und stellt die **Startbedingungen** her. Sie zieht die SDA-Leitung auf LOW-Pegel während die SCL-Leitung auf High-Pegel bleibt. Dies ist das **Aufweck-Signal** für alle Busbausteine! Alle Bausteine beobachten ("listen") nun die auf dem Bus anliegenden Daten.



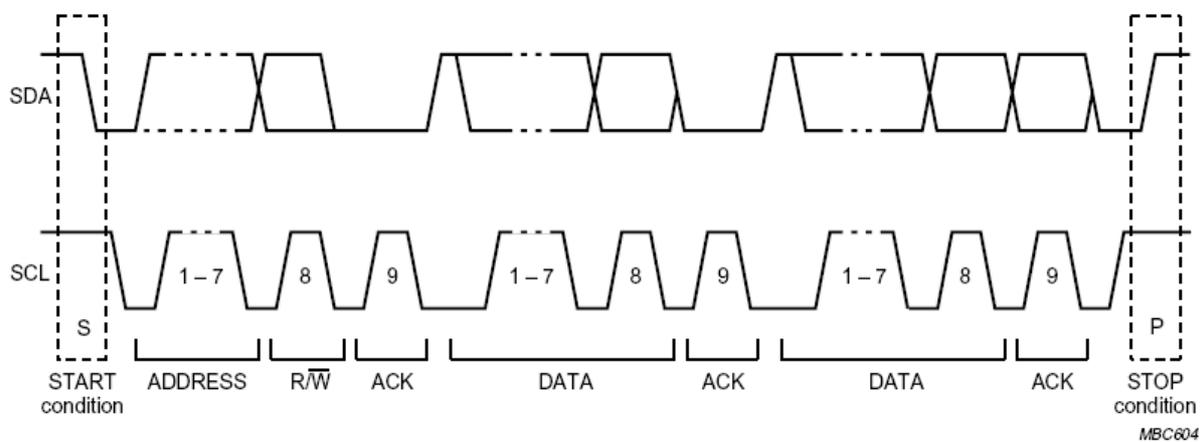
Generell quittiert der Empfänger von Daten den Empfang des Bytes mit einem **ACKNOWLEDGE-Bit A=0**. Dies erfolgt in der Weise, dass der Slave die



SDA-Leitung auf LOW-Pegel zieht und dies vom Master im neunten SCL-Takt überprüft wird. Ist der adressierte Baustein z.B. nicht vorhanden, so wird die SDA-Leitung

sich auf High-Pegel (Pull-Up-Widerstände !) befinden (**Not Acknowledge:**

**ACKNOWLEDGE-Bit A=1, SDA=high)** und der Master kann eine Fehlerbehandlung einleiten. Wurde die Übertragung vom Slave ordnungsgemäß quittiert, kann der Master weitere Daten byteweise an den Slave senden, wobei dieser den Empfang jedes Bytes mit ACKNOWLEDGE-Bit A=0 im neunten Shift-Clock-Takt bestätigt. Das Ende der Kommunikation wird durch die Herstellung der **STOP-Bedingung** erreicht. Sie ist dann gegeben, wenn die SDA-Leitung von LOW- auf HIGH-Pegel wechselt, während sich die SCL-Leitung auf High-Pegel befindet. **Aus diesem Grunde darf bei der normalen Datenübertragung nur dann Pegelwechsel auf SDA erfolgen, wenn SCL Low-Pegel führt:**



complete data transfer

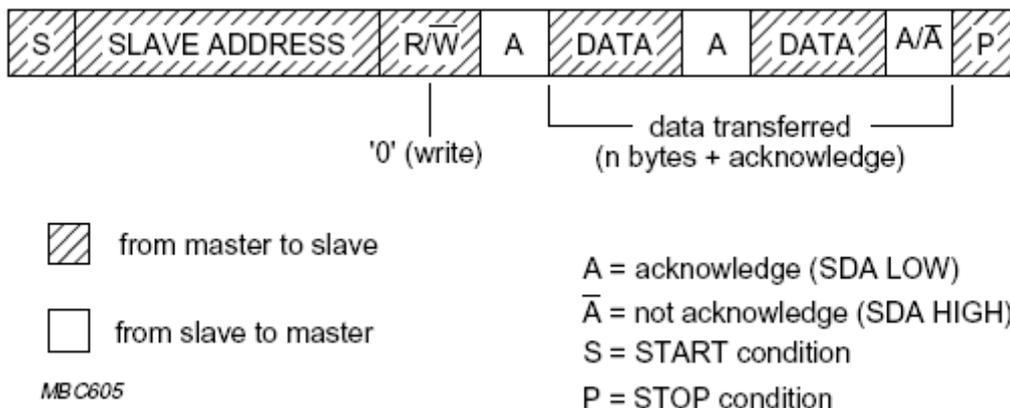
In der obigen Abbildung wird ein Baustein, etwa ein EEPROM (Basisadresse 1010XXX) auf dem Bus mit der Adresse 1010111 adressiert und ihm mitgeteilt, dass ein Auslesevorgang folgen soll. Das EEPROM hat die Datenübertragung verstanden und quittiert im neunten SCL-Takt mit einem LOW-Pegel auf der SDA-Leitung.

### 3.2 Die Datenübertragung zwischen Master und Slave

Hat der Master mit Startbedingung, Adressierung und R/W-Bit erst einmal einen Slave angesprochen, so kann in diesem Modus prinzipiell eine beliebige Zahl von Bytes vom Master zum Slave (R/W=0) oder vom Slave zum Master (R/W=1) übertragen werden. Dies erfolgt Byte-Weise, wobei der Empfänger (Slave oder Master) den Empfang mit einem LOW-Pegel auf der SDA-Leitung im neunten SCL-Puls bestätigt. Es sind folgende Fälle der Kommunikation zu unterscheiden;

- **Datenübertragung vom Master zum Slave**
- **Datenübertragung vom Slave zum Master**
- **Kombinierte Datenübertragung zwischen Master und Slave**

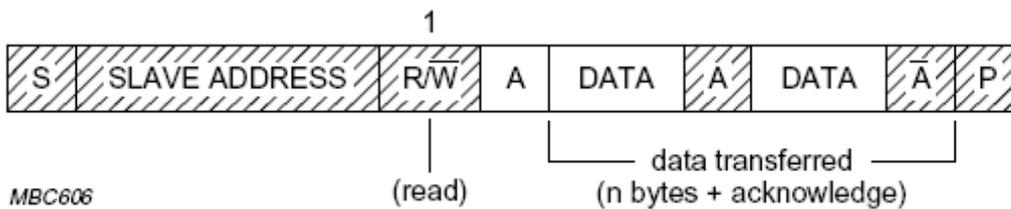
Wechselt der R/W-Modus so sind vom Master erneut die Startbedingungen wiederherzustellen und der entsprechende Baustein neu zu adressieren. Erst dann kann wieder die Datenkommunikation auf SDA aufgenommen werden. Die Fälle sollen nun i.E. betrachtet werden.



#### 3.2.1 Datenübertragung vom Master zum Slave

Der Master stellt zunächst die STARTBEDINGUNGEN her, indem er den SDA-Pegel von HIGH auf LOW zieht, während der SCL-Pegel HIGH bleibt. Danach wird der SCL-Pegel vom Master auf LOW gezogen und der Master taktet synchron mit der SCL-Leitung die 7-bit-Adresse des Slaves auf die SDA-Leitung. Mit dem Takten des achten Bits (LSB), des R/W-Bits=0, teilt er dem Slave mit, dass dieser die folgenden Bytes **empfangen** soll. Zwischen achtem

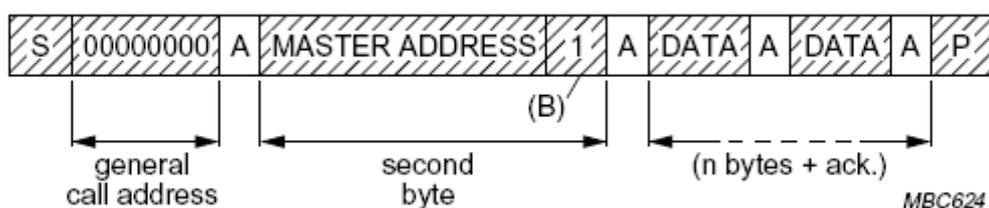
und neuntem SCL-Takt zieht der Slave die SDA-Leitung auf LOW-Pegel und bestätigt den Empfang des vom Master gesendeten Bytes (ACKNOWLEDGE=0). Nun kann der Master ein weiteres Byte zum Slave durchtakten, welches wiederum im neunten Takt durch einen LOW-Pegel aus SDA vom Slave bestätigt wird usw. Die Datenübertragung wird unterbrochen, wenn der Slave die Quittierung verweigert (ACKNOWLEDGE=1) und/oder der Master die STOPBEDINGUNGEN herstellt. Die folgende Abbildung verdeutlicht die Sequenz:



Datenübertragung vom Master zum Slave

### 3.2.2 Datenübertragung vom Slave zum Master

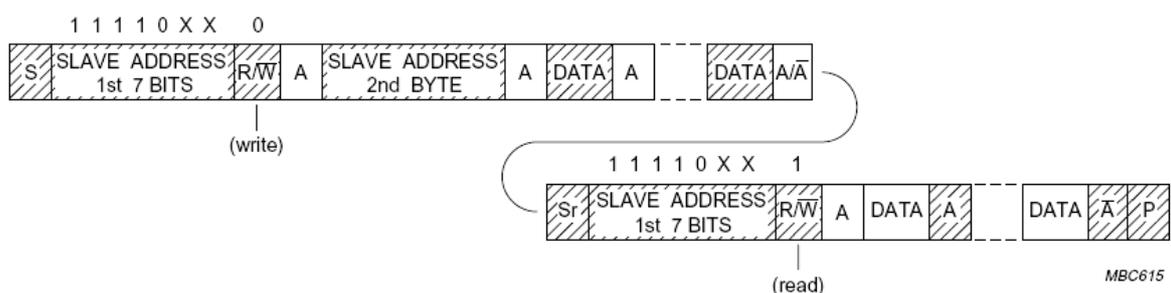
Auch hier stellt der Master zunächst die STARTBEDINGUNGEN her und Adressiert den anzusprechenden Baustein mit seiner 7bit-Adresse. Mit dem MSB (R/W=1) des vom Master auf den Bus gelegten Bytes teilt er dem Baustein (Slave) mit, dass Daten aus ihm ausgelesen werden sollen. Der Slave quittiert den empfang des Bytes, indem er zwischen dem achten und neunten Takt auf SCL die Datenleitung SDA auf LOW-Pegel zieht. Mit den nächsten acht Taktimpulsen des Masters auf der SCL-Leitung liefert der Slave 8-Bit Daten, die der Master empfängt. Er quittiert dies, indem er - zwischen achtem und neuntem Taktimpuls die SDA-Leitung auf LOW setzt (ACKNOWLEDGE=0). Dies ist das Signal für des Slave, ein weiteres 8-Bit-Datenpaket sequentiell auf de Bus zu legen, dessen Empfang wiederum vom Master mit einem LOW-Pegel auf SDA im neunten Takt von SCL bestätigt wird. Die Datenübertragung endet, wenn der Master das ACKNOWLEDGE verweigert (ACKNOWLEDGE=1, SDA=HIGH im neunten Taktimpuls) und die STOPBEDINGUNG einleitet.



Datenübertragung vom Slave zum Master

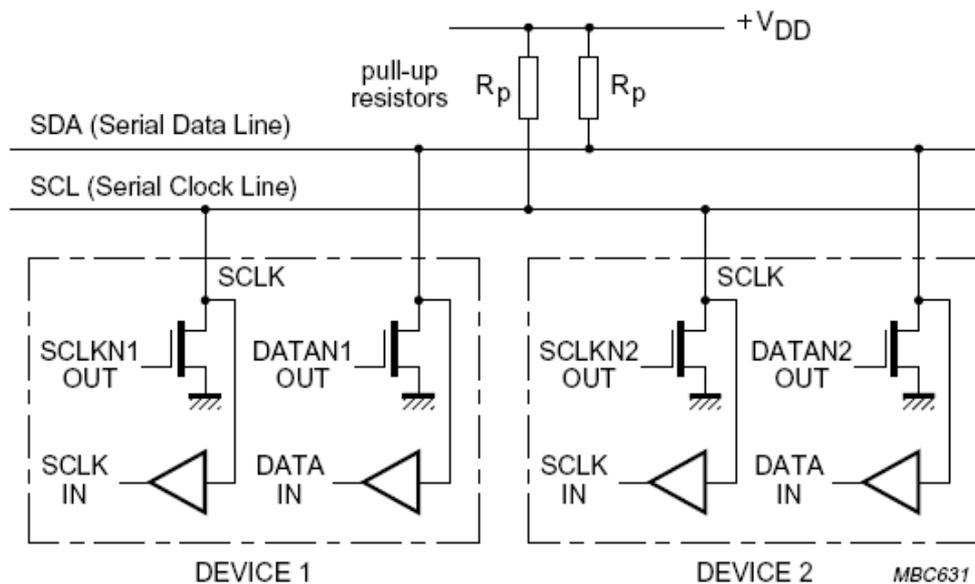
### 3.2.3 Kombinierte Datenübertragung zwischen Master und Slave

In diesem Fall stellt initial auch der Master zunächst die Starbedingungen her und wählt den Slave-Baustein mit 7 Adressbits an. Im achten Bit teilt der Master dem Slave mit, dass dieser nun Daten zu empfangen hat (R/W=0). Der Slave quittiert mit einem LOW-Pegel auf der SDA-Leitung den Empfang des Bytes, woraufhin der Master nach dem bekannten Muster (Master sendet 8 Bit, Slave quittiert mit LOW-Pegel auf der SDA-Leitung im neunten Taktimpuls auf der SCL-Leitung) weitere Daten an den Slave. Soll nun die Datenübertragungsrichtung umgekehrt werden, so muss der Master nach dem letzten ACKNOWLEDGE des Slaves erneut die Starbedingungen erzeugen (Start Recall, SR), die sieben Adressbits und das R/W-Bit (R/W=1, Read) senden - Der Slave quittiert nun mit SDA-Pegel LOW den Empfang des Bytes und sendet nun in den nächsten acht SCL-Takten die Daten zum Master. Im neunten Taktimpuls auf der SCL-Leitung signalisiert der Master dem Slave, dass er die Daten ordnungsgemäß empfangen hat. Daraufhin stellt der Slave weitere Datenbytes bitweise seriell auf SDA bereit die jeweils durch ein ACKNOWLEDGE (SDA=LOW) im neunten Taktimpuls auf SCL bestätigt werden. Beendet wird der Datentransfer durch die Verweigerung des ACKNOWLEDGE (SDA=HIGH) durch den Master und die Herstellung der STOPBEDINGUNG durch den Master (Die abgebildete Sequenz ist zeilenweise von links nach rechts hintereinander zu lesen!).



Kombinierte Datenübertragung zwischen Master und Slave

## 4.Installationstechnik



**Elektrische Schaltung**

I2C Bus wird fast nur auf die Leiterplatten oder zwischen Zwei Leiterplatten verwendet (Bei Modular Aufbau von Leiterplatten). Bsp. Zwischen Mikrocontroller LP und IR Empfänger LP.ein I2C Bus zurealisieren Man braucht nur Zwei Bus-Leitungen und Zwei Pullup Widerstände. Zusätzlich Serie Widerstände dienen als Überspannungsschutz.