

**ACHTUNG!** Diese Seite wird nicht mehr länger gepflegt.  
Wechselt auf die englischsprachige Seite für eine aktuelle  
Anleitung (Link unten links).

# Das I2C Master Protokoll

Der Netzer unterstützt I2C Master Emulation seit **Version 1.2**.

Ab Version 1.5 gibt es mehrere Möglichkeiten (Binds), wie Zeichen an die I2C-Schnittstelle gelangen können:

- [TCP/IP Socket](#)
- [Prozess](#)

Der größte Teil des Datentransfers wird vom Netzer transparent durchgereicht. Der Datentransfer ist zeichenorientiert, d.h. jedes zur I2C-Schnittstelle übertragene Zeichen (**Request**) wird bearbeitet, dann erfolgt die Antwort (**Response**) im Form eines Zeichens. Der Übertragungsrahmen im Sinne des I2C-Protokolls wird durch spezielle Zeichen signalisiert. Diese Zeichen erzeugen die speziellen I2C-Zustände am Bus wie START, STOP und REPEATED START.

## Das Terminierungsbyte

Das NULL Zeichen (ASCII 0x00) hat für die Kommunikation die spezielle Bedeutung eine laufende I2C Übertragung abubrechen. Dies gilt für beide Signalrichtungen:

- Wird an die I2C-Schnittstelle ein NULL gesendet, wird dort das STOP-Symbol erzeugt, was die Übertragung am I2C beendet.
- Antwortet die I2C-Schnittstelle mit einer NULL, ist auf dem Bus etwas schief gegangen (z.B. kein ACK durch adressierten Slave oder Bus ist belegt). In dem Fall wird automatisch ein STOP-Symbol auf dem Bus generiert.

Soll 0x00 auf dem I2C übertragen werden (nicht General call, s.u.), dann muss der Request mit vorangestelltem Escape-Zeichen (Backslash, ASCII 0x5C) übertragen werden. Gleichermassen muss für die Übertragung von 0x5C zuvor 0x5C übertragen werden.

## Die Startbedingung und das Adressierungsbyte

Ein START-Symbol wird automatisch auf dem Bus generiert, sobald das erste Zeichen an der I2C-Schnittstelle empfangen wird. Sollte dabei etwas schief gehen (z.B. Bus ist belegt), antwortet die

Schnittstelle mit 0x00.

Das I2C Protokoll definiert das erste Zeichen nach dem START-Symbol auf dem Bus als Slaveadresse. Sollte kein Slave antworten (ACK bit war '1'), antwortet die Schnittstelle mit 0x00. Das STOP-Symbol wird in dem Fall automatisch generiert.

Bei einer erfolgreichen Übertragung der Slaveadresse antwortet die I2C-Schnittstelle mit 0xFF.

## General call Adresse

Die I2C General call Adresse 0x00 adressiert alle Slaves am I2C Bus. 0x00 **muss** zu Beginn eines neuen I2C-Transfers **ohne** Escape übertragen werden.

## Das R/W Bit

Die Schnittstelle überprüft das R/W bit (LSB) der zuerst übertragenen Slaveadresse.

Ist das R/W bit **nicht gesetzt** wird eine Übertragung vom Master zum Slave gestartet. Jedes Byte was nun in Folge an die Schnittstelle gesendet wird, wird direkt über den I2C Bus übertragen (Escaping beachten). Als Antwort sendet die I2C-Schnittstelle entweder 0xFF für eine positive Bestätigung (ACK Bit war '0') oder 0x00 als negative Bestätigung (ACK Bit war '1'). Im Fall einer negativen Bestätigung wird automatisch das STOP-Symbol auf dem Bus generiert.

Ist das R/W bit **gesetzt** wird eine Übertragung vom Slave zum Master gestartet (Takt wird vom Master generiert). Nach der erfolgreichen Übertragung der Slaveadresse wartet die Schnittstelle auf weitere Bytes. Sobald ein Byte an die Schnittstelle übertragen wird, wird das nächste Byte vom Slave über I2C abgeholt und als Antwort gesendet. Sollte das Byte eine 0x00 sein, wird die Übertragung beendet und eine STOP-Bedingung generiert.

## Repeated Start

I2C unterstützt wiederholtes Übertragen ohne Busfreigabe. Dabei wird eine spezielle REPEATED START Bedingung auf dem Bus erzeugt, welche STOP und START kombiniert. Um REPEATED START zu erzeugen, muss das Symbol 's' (ASCII 0x73) als Request an die I2C-Schnittstelle übertragen werden. Soll 0x73 als Request übertragen werden, muss zuvor ein 0x5C übertragen werden. Das gilt nicht, wenn 0x73 als Slaveadresse übertragen wird (analog wie beim General Call). 0x73 in Antworten hat keine besondere Bedeutung.

## Beispiele

tbd

From:

<http://mobacon.de/wiki/> - MoBaCon Wiki

Permanent link:

<http://mobacon.de/wiki/doku.php/de/netzer/i2c-master>

Last update: **2014/02/09 17:00**

