The I2C master protocol

Netzer supports I2C master since version 1.2. The most data transactions are done transparently by Netzer. But a few protocol specifiers are implemented for handling the special states of I2C.

NULL - The termination character

The character *0x00* in the TCP/IP communication has special meaning in terminating any pending I2C transmission. If Netzer sends a *0x00* either the bus is busy, a bus collission occurred or a slave does not respond on address or data bytes. In the latter cases the bus is freed with a stop condition automatically. If Netzer receives *0x00* while a I2C transmission is pending, the transmission will be finished with a automatic generated stop condition.



If *0x00* should be sent via the I2C interface it must be escaped with a leading backspace. Therefore also the backslash itself must be escaped.

The start condition and the slave address byte

A start condition is automatically announced on the bus, if Netzer gets the first character via the TCP/IP serial socket. If an error occurs while announcing the start condition Netzer responds with a *0x00*. I2C protocol defines the very first byte after the start condition as slave address. Netzer sends a *0x00* via TCP/IP if no slave responds (ACK bit was '1'). Also a stop condition is initiated by Netzer. After the successful (means acknowledged) transmission of the slave address Netzer sends *0xFF* via TCP/IP.

General call address

The I2C general call address *0x00* addresses all slaves on the I2C bus. Sending *0x00* as the very first byte is allowed as an exception of the Netzer protocol. The escaping backslash MUST not be used.

The R/W bit

Netzer also checks the R/W bit (LSB) of the slave address character.

Master write

If the R/W bit is **cleared** a transmission from master to slave is initiated.

Each byte received by Netzer via the TCP/IP socket is directly sent to the slave (consider escaping!). In response Netzer sends via TCP/IP either '0xFF' as positive acknowledge (ACK bit was '0') or *0x00* as negative acknowledge (ACK bit was '1'). In case of a negative acknowledge Netzer initiates a stop condition on the bus.

Master read

If the R/W bit is **set** a transmission from slave to master is initiated (clock is generated by I2C master).

After the successful transmission of the slave address Netzer waits for further bytes on the TCP/IP socket. On receiving a byte via TCP/IP a byte is pulled from the slave and is sent via TCP/IP. In case the received byte is *0x00* the transmission is over and Netzer generates a stop condition on the bus. Any other value pulls the next byte from slave.

Repeated start

I2C supports repeated start. This symbol concatenates a stop and start condition while a I2C transmission is pending without freeing the bus inbetween.

The special character *s* (0x73) in TCP/IP stream initiates the repeated start on the bus. Therefore *0x73* must be escaped during normal stream transmission to Netzer. Exception to escaping *0x73* is the very first character (slave address) analog to the “General Call address” section.



Receiving *0x73* from Netzer has no special meaning but *0x73* and therefore will not be escaped!

From:
http://www.mobacon.de/dokuwiki/ - **MoBaCon**

Permanent link:
http://www.mobacon.de/dokuwiki/doku.php?id=en:netzer:i2c-master&rev=1419155800

Last update: **2025/06/11 20:42**