

SPI Master Ethernet Gateway

In diesem Modus können durch das Ethernet-Gateway Netzer direkt [SPI-Geräte](#) wie Speicher oder Displays angesprochen werden.

Dazu wird der Slave oder die Slaves am SPI-Interface des Netzers angeschlossen. Der Netzer fungiert dabei als SPI Master.

Durch die Natur des SPI werden Slaves zumeist über ein Chipselect ausgewählt. In der Regel also eine Leitung pro Slave. Manchmal werden auch noch andere Leitungen benötigt, um einen Slave steuern zu können - wie beispielsweise ein Resetsignal.

Auch die Schieberegister-Beschaltung ist möglich, dabei werden die Slaves wie in einer langen Kette hintereinander geschaltet (Chaining). Vorteil hierbei ist, dass sich alle Slaves eine Chipselect-Leitung teilen bzw. diese ganz entfällt. Das klassische Beispiel für Slaves im Chaining wäre das 8 Bit breite Schieberegister 74×594, welches beliebig oft hintereinander geschaltet werden kann.

Einstellungen am Webinterface

Der SPI Master Modus wird auf der allgemeinen Einstellungsseite aktiviert.



The screenshot shows a web interface with a dark red background. The settings are as follows:

- Netzwerkname: ne2c3d9f.local
- HTTP Port: 80
- GPIO TCP Port: 65000 ☒ Aktiv
- Serieller Modus: A dropdown menu is open, showing options: Deaktiviert, UART, SPI Master (highlighted), SPI Slave, and I2C Master.
- Serieller Port: (This field is partially obscured by the dropdown menu)

Below the settings, there is a yellow text warning: "Änderungen werden erst nach Neustart wirksam". At the bottom, there are two buttons: "Speichern" and "Zurücksetzen".

Nach Änderung des Modes Neustart nicht vergessen!

Für TCP-Verbindungen wird der Port des seriellen Servers verwendet.

Desweiteren werden SPI-Einstellungen auf der Einstellungsseite für die seriellen Schnittstellen vorgenommen.

STATUS

ALLGEMEIN

GPIO

SERIELL

UART Einstellungen

UART Baudrate:

19200 Baud

Allgemeine SPI Einstellungen (SPI Slave: Änderungen erst nach Neustart)

Taktmodus:

☒ SPI Mode 0
☐ SPI Mode 1
☐ SPI Mode 2
☐ SPI Mode 3

SPI Master Einstellungen

Taktfrequenz:

10.4 MBit/s

Abtastung des Eingangs:

☒ In der Mitte
☐ Am Ende

I2C Master Einstellungen (SCL = SPI_CLK, SDA = SPI_MI)

Taktfrequenz:

400 kHz

Speichern

Zurücksetzen

Es können die folgenden Frequenzen für SPI_CLK eingestellt werden:

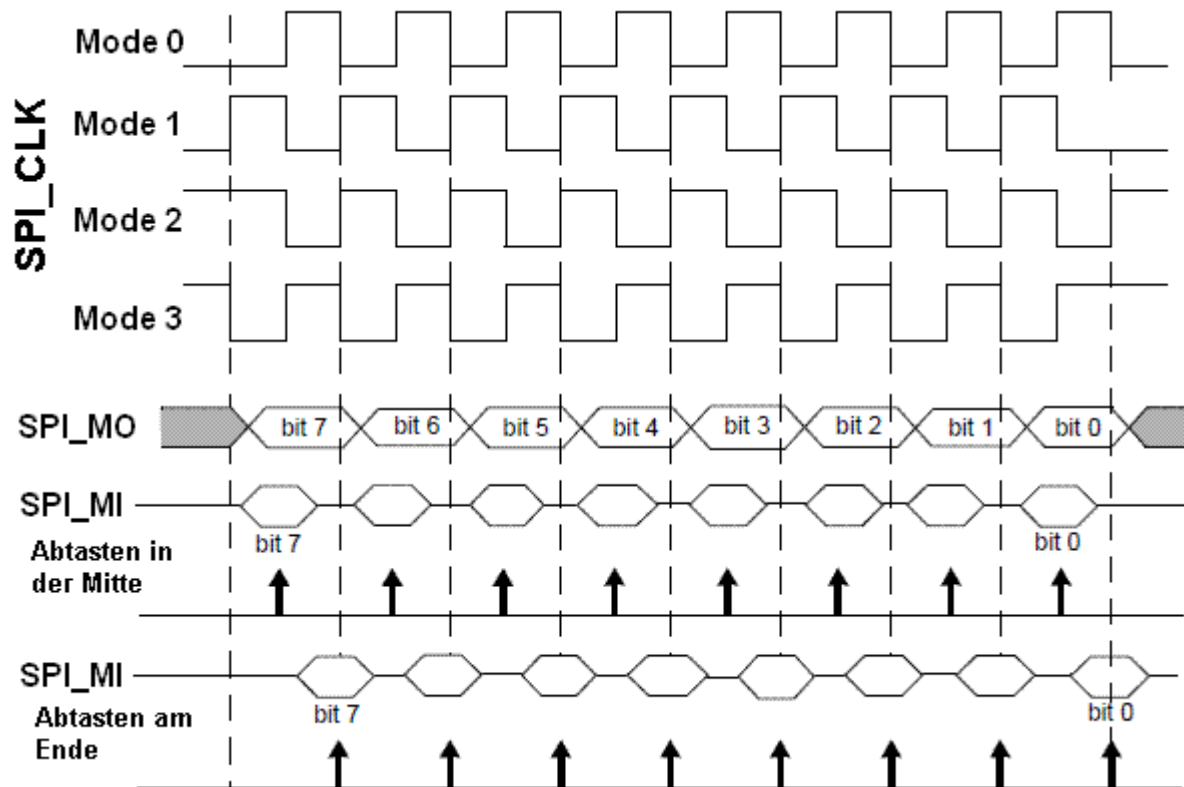
Frequenz	Ab Firmware-Version
10,4 MBit/s (Wow!)	1.0
2,6 MBit/s	1.0
650 kBit/s	1.0
100 kBit/s	1.5
50 kBit/s	1.5
10 kBit/s	1.5
5 kBit/s	1.5
1,3 kBit/s	1.5

Für SPI ist insbesondere der Taktmodus wichtig. Er bestimmt die Polarität des Taktsignals sowie die Übernahmeflanke wie folgt:

Modus	Ruhezustand des Taktsignals	Übernahmeflanke
Modus 0	0	Steigend
Modus 1	0	Fallend
Modus 2	1	Fallend
Modus 3	1	Steigend

Für die erfolgreiche Übertragung von Daten müssen die Modi bei Master und Slave(s) gleich sein!

Die Grafik zeigt alle Einstellungen im Überblick anhand eines Zeitdiagramms:

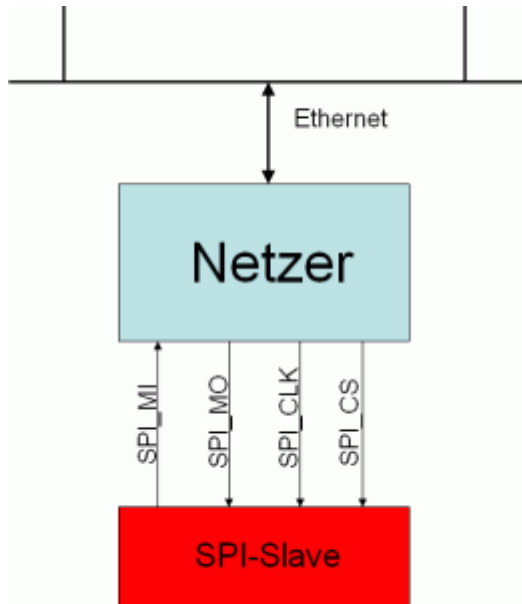


Kommunikation

Sobald eine [Verbindung](#) aufgebaut ist, kann der SPI Master verwendet werden. Alle vom TCP Client empfangenen Zeichen bis auf den Backslash '\' (Escape) werden durch den Netzer transparent an das SPI Master Modul durchgeleitet. Escaped Zeichen, die von der jeweiligen Firmwareversion nicht unterstützt werden, werden ebenfalls transparent durchgeleitet. Für eine vollständige Auflistung der unterstützten Escape-Sequenzen siehe weiter unter. Der 256 Byte große Ringpuffer auf Netzerseite puffert größere Datenbursts.

Das SPI Master Modul generiert für jedes Byte 8 Takte an SPI_CLK und schiebt das Datum an SPI_MO hinaus. Zeitgleich liefert der Slave im Takt des Masters Daten an SPI_MI (siehe Grafik oben). Das so empfangene Byte wird automatisch an den TCP Client gesendet. In diese Richtung wird kein Escaping verwendet.

Firmware Basisversion



Mit der frei downloadbaren Basisversion wird genau ein SPI Slave unterstützt bzw. mehrere SPI Slaves im Chaining, da dafür nur ein Chipselect benötigt wird. Eine Beispielbeschaltung wie im Bild oben besteht aus den vier Pins SPI_MI, SPI_MO, SPI_CLK sowie SPI_CS. SPI_MO ist der Datenausgang, d.h. Daten vom Netzer zum Slave. SPI_MI ist der Dateneingang, Daten vom Slave zum Netzer. Die Taktleitung (SPI_CLK) wird vom Netzer gesteuert.

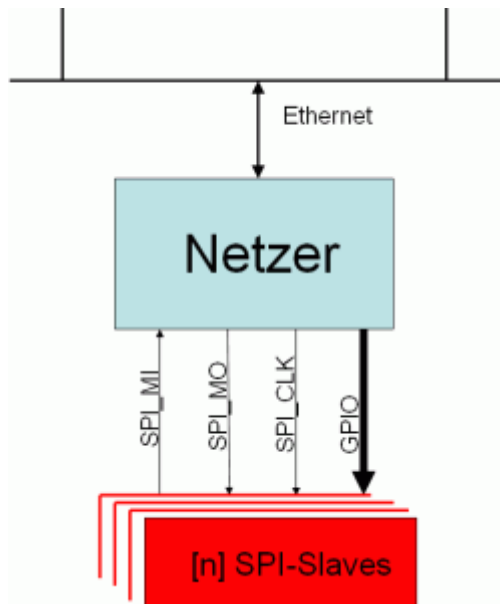
SPI_CS dient der Selektierung und Synchronisierung des Slaves - sobald Daten übertragen werden, setzt der Netzer diese Leitung automatisch auf 0. Das Zurücksetzen auf 1 erfolgt durch eine spezielle Escapesequenz (0x5C 0x00).

Alle anderen GPIO-Pins des Netzers sind wie gewohnt frei verwendbar und z.B. über das Webinterface bzw. den [GPIO-Server](#) steuerbar.

Unterstützte Escape-Sequenzen in der Basisversion:

Sequenz (in hex)	Bedeutung
0x00	SPI_CS wird zurückgesetzt auf 1 - die aktuelle Slavesession wird damit beendet.
Alle anderen	Zeichen wird transparent durchgereicht.

Firmware Proversion



Im Unterschied zur Basisversion nutzt die Proversion keine dedizierte ChipSelect-Leitung sondern bietet über spezielle Escapesequenzen Zugriff auf alle Netzer-Pins, die als digitale Ausgänge konfiguriert sind. Damit können nicht nur bis zu 10 verschiedene SPI-Slaves selektiert werden. Es können damit auch spezielle Steuerepins bedient werden, die bei speziellen SPI Bausteinen benötigt werden.

Alle GPIO-Pins (bis auf SPI_CLK, SPI_MO und SPI_MI) des Netzers sind auch frei verwendbar und z.B. über das Webinterface bzw. den [GPIO-Server](#) steuerbar.

Weiterhin kann der SPI-Mode dynamisch umkonfiguriert werden. Ferner können Wartezeiten eingefügt werden, um z.B. bestimmte Setup/Hold-Zeiten von Slaves zu erreichen.

Um den Netzer nicht zu lang zu blockieren, werden alle empfangenen Escape-Sequenzen sequentiell bearbeitet. D.h. zwischen der Verarbeitung von Escape-Sequenzen entstehen mehr oder weniger große Pausen, was zu leichten Jitter von Kontrollsequenzen führen kann. Um feste Zeiten zwischen Signalwechseln zu erreichen, sollten dafür die Escapesequenzen für Wartezeiten verwendet werden.

Unterstützte Escape-Sequenzen in der Proversion:

Sequenz (in ASCII)	Sequenz (in hex)	Bedeutung	Ab Firmware-Version
0	0x30	SPI-Mode 0 wird aktiviert.	1.5
1	0x31	SPI-Mode 1 wird aktiviert.	1.5
2	0x32	SPI-Mode 2 wird aktiviert.	1.5
3	0x33	SPI-Mode 3 wird aktiviert.	1.5
a	0x61	Setze Pin IO0 auf 0.	1.4
b	0x62	Setze Pin IO1 auf 0.	1.4
c	0x63	Setze Pin IO2 auf 0.	1.4
d	0x64	Setze Pin IO3 auf 0.	1.4
e	0x65	Setze Pin IO4 auf 0.	1.4
f	0x66	Setze Pin IO5 auf 0.	1.4
g	0x67	Setze Pin TX auf 0.	1.4
h	0x68	Setze Pin RX auf 0.	1.4

i	0x69	Setze Pin SPI_CS auf 0.	1.4
j	0x6A	Setze Pin SPI_INT auf 0.	1.4
A	0x41	Setze Pin IO0 auf 1.	1.4
B	0x42	Setze Pin IO1 auf 1.	1.4
C	0x43	Setze Pin IO2 auf 1.	1.4
D	0x44	Setze Pin IO3 auf 1.	1.4
E	0x45	Setze Pin IO4 auf 1.	1.4
F	0x46	Setze Pin IO5 auf 1.	1.4
G	0x47	Setze Pin TX auf 1.	1.4
H	0x48	Setze Pin RX auf 1.	1.4
I	0x49	Setze Pin SPI_CS auf 1.	1.4
J	0x4A	Setze Pin SPI_INT auf 1.	1.4
t	0x74	Warte 100 µs bis zur Verarbeitung des nächsten Zeichens.	1.5
T	0x54	Warte 1 ms bis zur Verarbeitung des nächsten Zeichens.	1.5
Alle anderen		Zeichen wird transparent durchgereicht.	1.5

From:

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

Permanent link:

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

Last update: **2014/02/09 13:32**

