

# SPI – Library

A.Schultze, DK4AQ, 10.06.2013

- ★ **Einleitung**
- ★ **Übersicht**
- ★ **Beschreibung**
  - **SPI.begin()**
  - **SPI.end()**
  - **SPI.setBitOrder()**
  - **SPI.setClockDevider()**
  - **SPI.setDataMode()**
  - **SPI.transfer()**

# SPI – Library

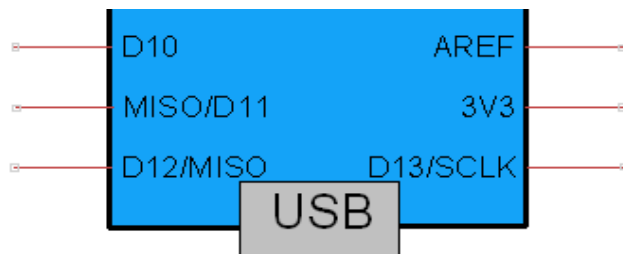
Nach <http://arduino.cc/en/Reference/SPI>

## Einleitung

Die SPI-Library gehört zum Standardumfang der Arduino-Libraries. Sie wird verwendet um die Hardware des ATmega328 zu initialisieren und einen einfachen Zugriff auf die Schnittstelle zu ermöglichen. Die Library arbeitet standardmässig mit 4Mb/s. Es wird nur der Master-Betrieb unterstützt !

Die SPI-Library arbeitet mit den Signalen / Pins:

Anschluss	MOSI	MISO	SCK	SS (als Slave)	SS (als Master)
Ard.Nano	D11	D12	D13	10	Dx
ICSP	4	1	3	-	-



# SPI – Library

## Übersicht

<b>SPI.begin()</b>	Initialisiert den SPI-Bus durch setzen der Pins SCK. MOSI als Ausgang, zieht SCK and MOSI auf den Pegel LOW
<b>SPI.end()</b>	Disable des SPI-Bus. Die Ausmodi der Pins werden nicht verändert
<b>SPI.setBitOrder(order)</b>	Setzt die Reihenfolge der auszugebenden Bits (LSB/MSB zuerst)
<b>SPI.setClockDivider(divider)</b>	Einstellung der Clockfrequenz, Teilung der Quarzfrequenz durch 2er-Potenzen
<b>SPI.setDataMode(mode)</b>	Einstellung des Clock-Modus (CPOL,CPHA)
<b>SPI.transfer(val)</b>	Sendet 1 Byte (val) auf den Bus und empfängt 1 Byte (Rückgabewert)

# SPI – Library

## Beschreibung

### **SPI.begin()**

Initialisiert den SPI-Bus durch Initialisierung der Pins SCK, MOSI und SS als Ausgang, zieht SCK and MOSI auf den Pegel LOW.

SS (Slave Select) ist der Pin D10 und wird in dieser Library nicht benutzt, da hier nur der Master-Betrieb realisiert wird.

### **Übergabeparameter:**

keine

### **Rückgabeparameter:**

keine

# SPI – Library

## **SPI.end()**

Sperrung (disable) der SPI-Schnittstelle. Die Initialisierung der Ausgangspins wird nicht verändert.

### **Übergabeparameter:**

keine

### **Rückgabeparameter:**

keine

## **SPI.setBitOrder(order)**

Setzt die Reihenfolge der auszugebenden Bits (LSB/MSB zuerst)

### **Übergabeparameter:**

LSBFIRST oder MSBFIRST

### **Rückgabeparameter:**

keine

# SPI – Library

## SPI.setClockDivider(divider)

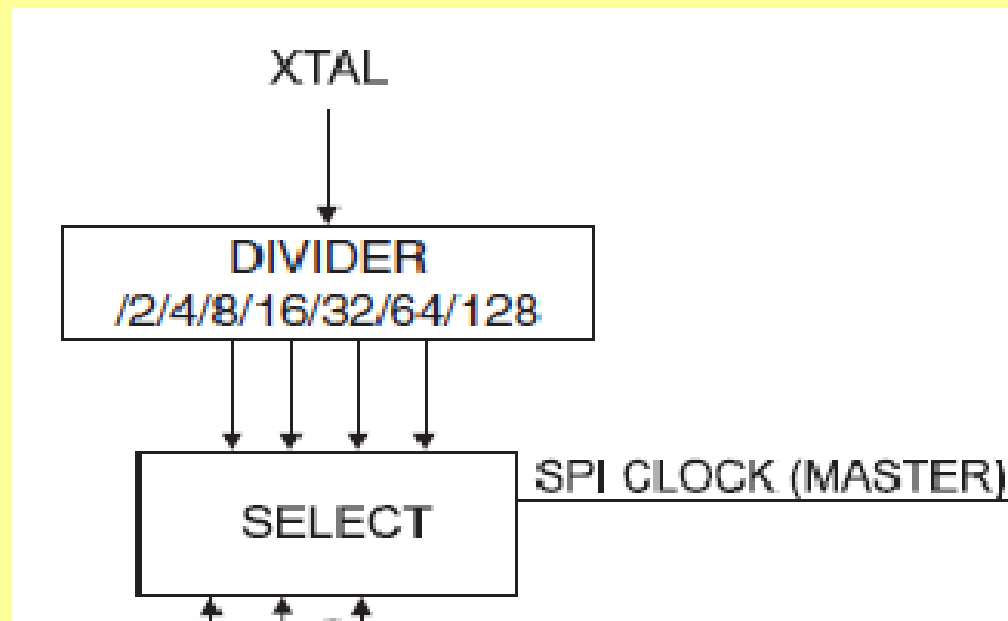
Einstellung der Clockfrequenz, Teilung der Quarzfrequ. durch 2er-Potenzen

### Übergabeparameter:

SPI\_CLOCK\_DIV2  
SPI\_CLOCK\_DIV4  
SPI\_CLOCK\_DIV8  
SPI\_CLOCK\_DIV16  
SPI\_CLOCK\_DIV32  
SPI\_CLOCK\_DIV64  
SPI\_CLOCK\_DIV128

### Rückgabeparameter:

keine



# SPI – Library

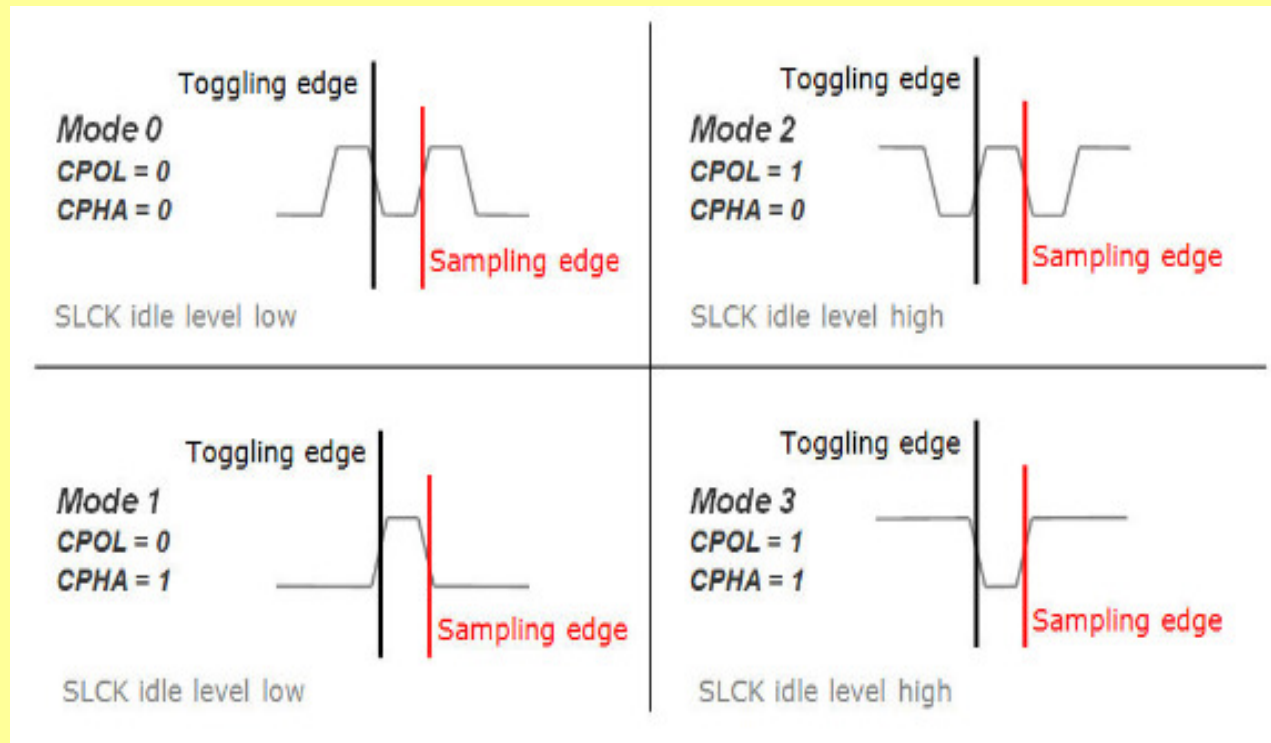
## SPI.setDataMode(mode)

Einstellung des Clock-Modus (CPOL,CPHA)

### Übergabeparameter:

SPI\_MODE0  
SPI\_MODE1  
SPI\_MODE2  
SPI\_MODE3

Rückgabeparameter:  
keine



# SPI – Library

## **SPI.transfer(val)**

Sendet 1 Byte auf den Bus über MOSI und empfängt 1 Byte über MISO. Ein Chip-Select-Signal wird nicht beeinflusst und muss bei Bedarf über ein Portpin vor und nach Senden gesetzt werden. Bei Übertragung mehrerer Bytes muss SPI.transfer in einer Schleife betrieben werden. Das gilt sowohl für die sendebytes als auch für die abzuholenden Empfangsbytes.

Die Bedeutung der Bytes ist nicht in SPI definiert, sondern muss aus dem Datenblatt des anzuschliessenden Bausteins ermittelt werden.

### **Übergabeparameter:**

Zu sendendes Byte

### **Rückgabeparameter:**

Empfangenes Byte