

Vortrag Projektlabor
Von Sophia Bauknecht
13.05.2013

Serial Peripheral Interface

SPI

Inhaltsverzeichnis

- Was ist SPI?
- Wozu braucht man SPI?
- Wie funktioniert SPI?
- Wie benutzt man SPI?
- Einstellungsmöglichkeiten
- Beispiel
- Vorteile
- Nachteile
- Einsatzbeispiele
- Weiterentwicklung
- Quellen

Was ist SPI?

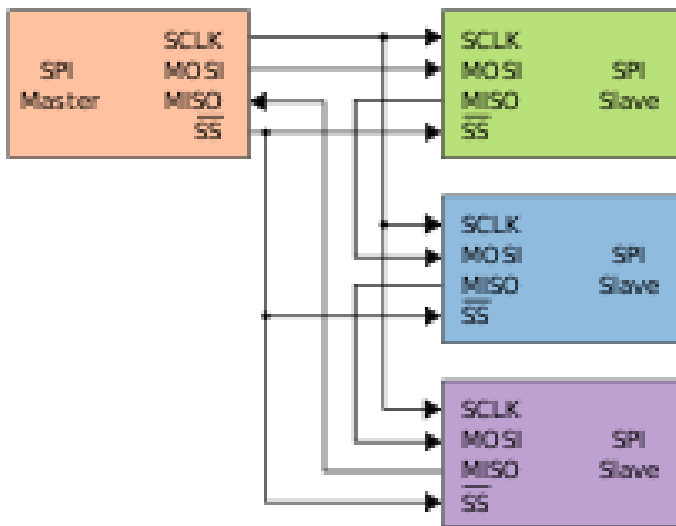
- Ein Bus-System bestehend aus drei Leitungen
- Zur seriell synchronen Datenübertragung zwischen Master und Slave

Wozu braucht man SPI?

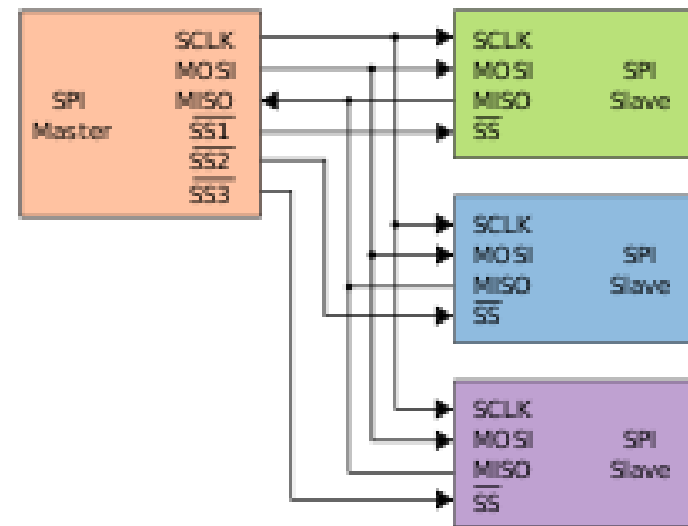
- Datentransfer zwischen 2 Systemen
- Meist zwischen der CPU und der Peripherie
aber auch zwischen 2 Microcontrollern

Wie funktioniert SPI?

- sehr lockerer Standard (sehr flexibel)
- digitale Schaltungen nach Master-Slave-Prinzip



Kaskadierung der Slaves



Sternverbindung

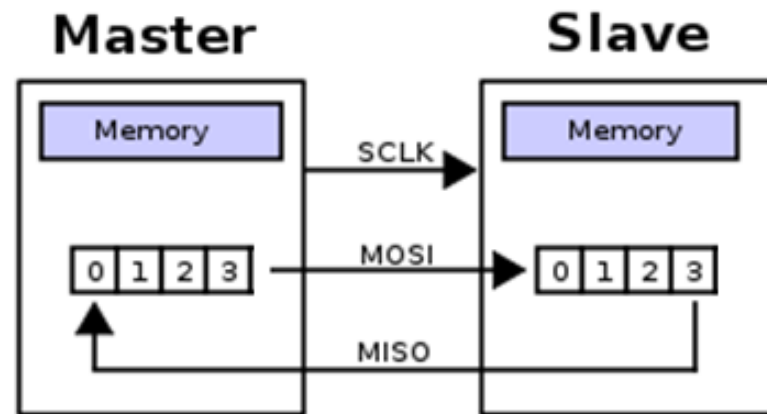
Wie funktioniert SPI?

- Beliebig viele Teilnehmer
- Aber nur ein Master

- Alle Bauteile auf der gleichen Platine
- Für Anbindung von externen Komponenten ungeeignet
- Leitungen werden vom Master gesteuert

Wie benutzt man SPI?

- Kommunikationstarten:
 - Frequenz der clock einstellen
 - Logische Null über die chip select Leitung
 - Starten der clock
- Datenübertragung:
 - Ein Bit pro Taktperiode
 - Master sendet auf MOSI
 - Slave sendet auf MISO



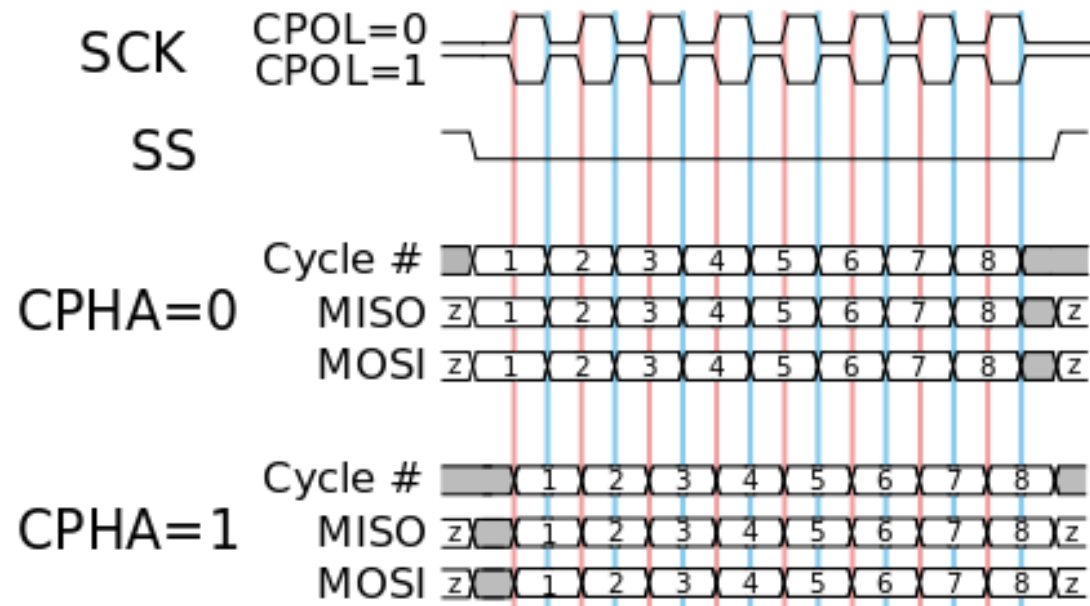
Wie benutzt man SPI?

- Beenden:
 - wenn das Slave-Select-Signal endgültig auf High gesetzt wird

Warum gibt es verschiedene Einstellungsmöglichkeiten?

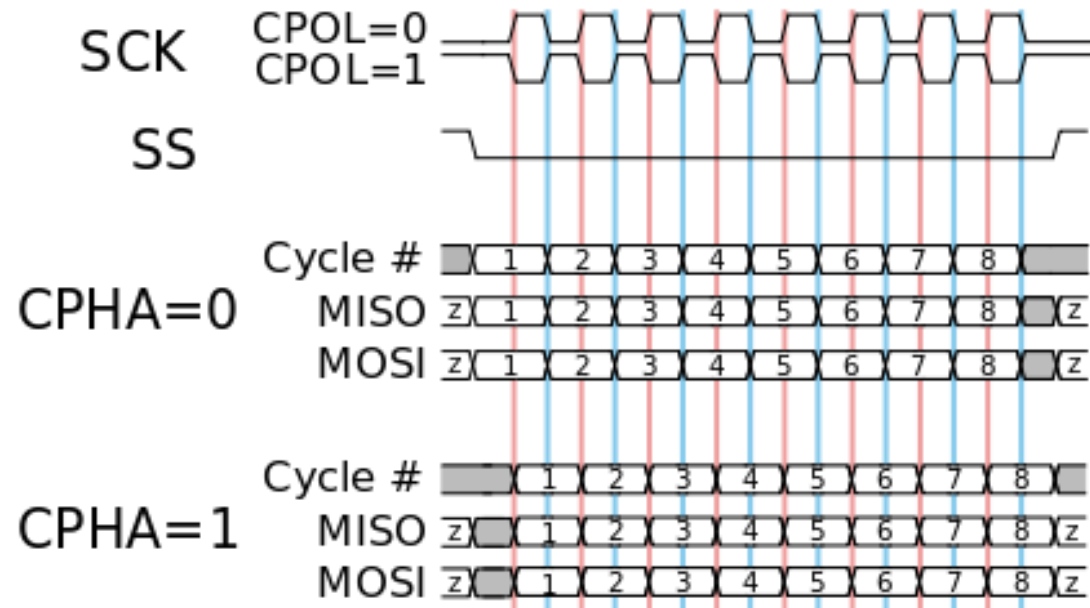
- Die Spezifikation nicht konkret festlegt
- Existenz verschiedener, zueinander inkompatibler Geräte

Datenübertragung bei verschiedenen Einstellungen von CPOL und CPHA



Ein Beispiel

```
unsigned char SPIWriteData(unsigned char byte){  
    unsigned char bit;  
    /* Die clock wird mit Null aktiviert*/  
    SETGPOL(o);  
    /* Datenübertragung auf der fallenden Flanke*/  
    SETGPHA(o);
```



Ein Beispiel

```
/* ein Byte soll übertragen werden*/
for (bit = 0; bit < 8; bit++) {
    /* clock Verzögerung vor der steigenden Flanke*/
    SPIDELAY(SPISPEED/2);
    SETCLK();
    /* gclock Verzögerung vor der fallenden Flanke */
    SPIDELAY(SPISPEED - (SPISPEED/2));
    CLRCLK();
    /* solange die 8 Bit nicht vorbei sind MOSI setzen*/
    if (byte & 0x80){
        SETMOSI();
    }
    else{
        CLRMOSI();
        byte <<= 1;
    }
}
return byte;
}
```

Vorteile

- hohe Datenübertragungsraten
- Übertragung in beide Richtungen gleichzeitig
- Einfache Verdrahtung
- Geringe Störraten zwischen den Kabeln
- Sehr hohe Programmflexibilität
- Sehr übersichtliche Hardware

Nachteile

- braucht mehr Pins als alternativen zu SPI
- Keine Transferkontrolle durch den Slave
- Befehle nur von einem Masker akzeptiert

Einsatzbeispiele

- Für Sensoren aller Art
 - Temperatursensoren
 - Tastsensoren
- Kammerlinsen
- Videocontrollern
- Memorykarten
- Echtzeit Uhren

Weiterentwicklung

- Weiterentwicklung durch Motorola
- QSPI enthält Warteschlange
- effektiveres Softwaredesign

Quellen

- <http://www.uni-koblenz.de/~physik/informatik/MCU/SPI.pdf>
- http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
- http://de.wikipedia.org/wiki/Serial_Peripheral_Interface
- <http://whatis.techtarget.com/definition/serial-peripheral-interface-SPI>
- http://www.mikrocontroller.net/articles/Serial_Peripheral_Interface