

MIKROKONTROLLER & I²C BUS



by AS

www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial

SPI mit dem Attiny 841
Teil 1 - Grundlagen

SPI und der
Attiny 841



Copyright

Sofern nicht anders angegeben, stehen die Inhalte dieser Dokumentation unter einer „Creative Commons - Namensnennung-NichtKommerziell-Weitergabe unter gleichen Bedingungen 3.0 DE Lizenz“



Sicherheitshinweise

Lesen Sie diese *Gebrauchsanleitung*, bevor Sie diesen Bausatz in Betrieb nehmen und bewahren Sie diese an einem für alle Benutzer jederzeit zugänglichen Platz auf. Bei Schäden, die durch Nichtbeachtung dieser Bedienungsanleitung verursacht werden, erlischt die *Gewährleistung / Garantie*. Für Folgeschäden übernehmen wir keine Haftung! Bei allen Geräten, die zu ihrem Betrieb eine elektrische Spannung benötigen, müssen die gültigen VDE-Vorschriften beachtet werden. Besonders relevant sind für diesen Bausatz die VDE-Richtlinien VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860. Bitte beachten Sie auch nachfolgende Sicherheitshinweise:

- Nehmen Sie diesen Bausatz nur dann in Betrieb, wenn er zuvor berührungssicher in ein Gehäuse eingebaut wurde. Erst danach darf dieser an eine Spannungsversorgung angeschlossen werden.
- Lassen Sie Geräte, die mit einer Versorgungsspannung größer als 24 V- betrieben werden, nur durch eine fachkundige Person anschließen.
- In Schulen, Ausbildungseinrichtungen, Hobby- und Selbsthilfwerkstätten ist das Betreiben dieser Baugruppe durch geschultes Personal verantwortlich zu überwachen.
- In einer Umgebung in der brennbare Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können, darf diese Baugruppe nicht betrieben werden.
- Im Falle einer Reparatur dieser Baugruppe, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen. Eine Reparatur des Gerätes darf nur von fachkundigen Personen durchgeführt werden.
- Spannungsführende Teile an dieser Baugruppe dürfen nur dann berührt werden (gilt auch für Werkzeuge, Messinstrumente o.ä.), wenn sichergestellt ist, dass die Baugruppe von der Versorgungsspannung getrennt wurde und elektrische Ladungen, die in den in der Baugruppe befindlichen Bauteilen gespeichert sind, vorher entladen wurden.
- Sind Messungen bei geöffnetem Gehäuse unumgänglich, muss ein Trenntrafo zur Spannungsversorgung verwendet werden
- Spannungsführende Kabel oder Leitungen, mit denen die Baugruppe verbunden ist, müssen immer auf Isolationsfehler oder Bruchstellen kontrolliert werden. Bei einem Fehler muss das Gerät unverzüglich ausser Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- Es ist auf die genaue Einhaltung der genannten Kenndaten der Baugruppe und der in der Baugruppe verwendeten Bauteile zu achten. Gehen diese aus der beiliegenden Beschreibung nicht hervor, so ist eine fachkundige Person hinzuzuziehen

Bestimmungsgemäße Verwendung

- Auf keinen Fall darf 230 V~ Netzspannung angeschlossen werden. Es besteht dann Lebensgefahr!
- Dieser Bausatz ist nur zum Einsatz unter Lern- und Laborbedingungen konzipiert worden. Er ist nicht geeignet, reale Steuerungsaufgaben jeglicher Art zu übernehmen. Ein anderer Einsatz als angegeben ist nicht zulässig!
- Der Bausatz ist nur für den Gebrauch in trockenen und sauberen Räumen bestimmt.
- Wird dieser Bausatz nicht bestimmungsgemäß eingesetzt kann er beschädigt werden, was mit Gefahren, wie z.B. Kurzschluss, Brand, elektrischer Schlag etc. verbunden ist. Der Bausatz darf nicht geändert bzw. umgebaut werden!
- Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller, sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und /oder Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.
- Der Autor dieses Tutorials übernimmt keine Haftung für Schäden. Die Nutzung der Hard- und Software erfolgt auf eigenes Risiko.

Attiny 841 und SPI - Teil 1 - Grundlagen

Gliederung

1. Einführung
2. Wie funktioniert SPI?
3. SPI-Hardware
4. SPI-Betriebsmodi
 - 4.1. Modus 0
 - 4.2. Modus 1
 - 4.3. Modus 2
 - 4.4. Modus 3
5. SPI-Konfigurationen
6. Anwendungen von SPI
7. Vorteile
8. Nachteile

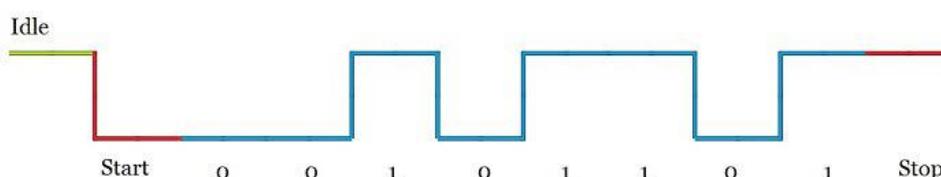
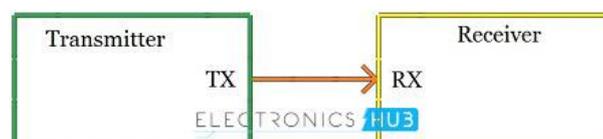
1. Einführung

Serial Peripheral Interface oder SPI ist ein synchrones serielles Kommunikationsprotokoll, das eine Vollduplex-Kommunikation mit sehr hohen Geschwindigkeiten ermöglicht. Serial Peripheral Interface (SPI) ist ein Master-Slave-Protokoll, das eine einfache und kostengünstige Schnittstelle zwischen einem Mikrocontroller und seinen Peripheriegeräten bietet.

Der SPI-Schnittstellenbus wird üblicherweise für die Verbindung von Mikroprozessoren oder Mikrocontrollern mit Speichern wie EEPROM, RTC (Real Time Clock), ADC (Analog-Digital-Wandler), DAC (Digital-Analog-Wandler), Displays wie LCDs, Audio-ICs, Sensoren wie Temperatur und Druck, Speicherkarten wie MMC- oder SD-Karten oder sogar anderen Mikrocontrollern verwendet.

Bei UART (oder jeder anderen üblichen seriellen Schnittstelle), bei der die Kommunikation über die RX- und TX-Leitung erfolgt, gibt es kein Taktsignal, d.h. es handelt sich um eine asynchrone Kommunikation. Bei dieser Art der Kommunikation gibt es keine Kontrolle über die gesendeten Daten oder darüber, ob Sender und Empfänger die gleiche Datenrate haben.

Um dies zu vermeiden, verwendet UART Synchronisationsbits, d. h. Startbit und Stopbits, sowie eine vorher vereinbarte Datenübertragungsgeschwindigkeit (normalerweise 9600 bps). Wenn die Baudraten von Sender und Empfänger nicht übereinstimmen, kommen die vom Sender

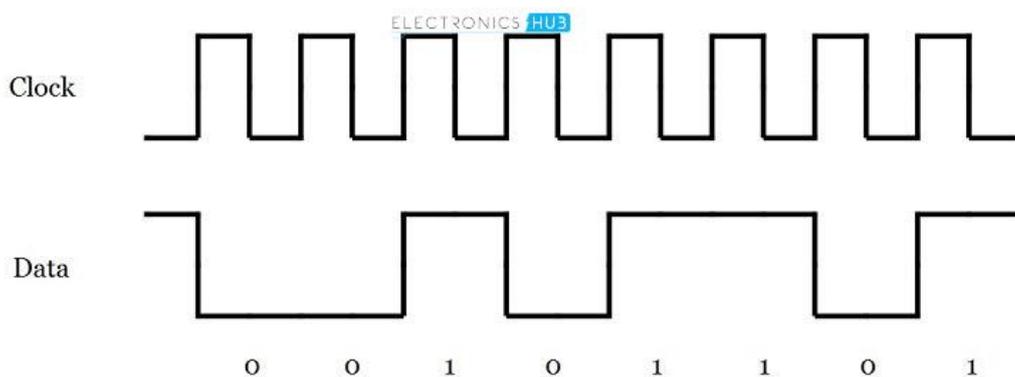
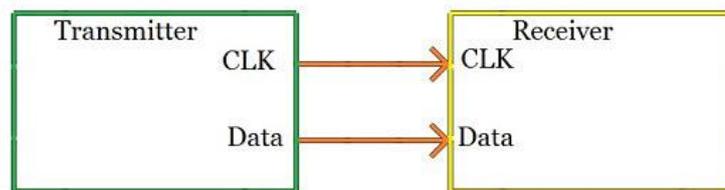


gesendeten Daten nicht richtig beim Empfänger an, und oft werden Müll- oder Junk-Werte empfangen.

Für die Kommunikation über kurze Entfernungen ist die synchrone serielle Kommunikation die bessere Wahl, insbesondere die serielle periphere Schnittstelle oder SPI. Wenn wir von Kurzstreckenkommunikation sprechen, ist damit oft die Kommunikation innerhalb eines Geräts oder zwischen den Geräten auf derselben Platine gemeint.

Die andere Art des synchronen seriellen Kommunikationsprotokolls ist I²C (Inter - Integrated Communication, oft auch als I Squared C oder I Two C bezeichnet). In diesem Artikel werden wir uns auf SPI konzentrieren.

SPI ist eine synchrone serielle Kommunikation, d. h. es verwendet ein spezielles Taktsignal zur Synchronisierung von Sender und Empfänger bzw. Master und Slave (in der SPI-Sprache). Der Sender und der Empfänger sind mit separaten Daten- und Taktleitungen verbunden, und das Taktsignal hilft dem Empfänger, wann er nach Daten auf dem Bus suchen muss.



Das Taktsignal muss vom Master an den Slave geliefert werden (oder an alle Slaves im Falle einer Mehrfach-Slave-Einrichtung). Es gibt zwei Arten von Triggerungsmechanismen für das Taktsignal, die verwendet werden, um dem Empfänger die Daten mitzuteilen:

- Flanken-Triggerung
- Pegel-Triggerung.

Die am häufigsten verwendete Triggerung ist die Flankentriggerung, und es gibt zwei Arten:

- steigende Flanke (Übergang von Low zu High am Taktsignal)
- fallende Flanke (Übergang von High zu Low). Je nachdem, wie der Empfänger konfiguriert ist, sucht er nach der Erkennung der Flanke nach Daten auf dem Datenbus des nächsten Bits

Da sowohl der Takt als auch die Daten vom Master (oder Sender) gesendet werden, brauchen wir uns keine Gedanken über die Geschwindigkeit der Datenübertragung zu machen.

Was SPI unter den anderen Protokollen der synchronen seriellen Kommunikation (oder jeder anderen seriellen Kommunikation) so beliebt macht, ist die Tatsache, dass es eine gesicherte Hochgeschwindigkeitsdatenübertragung mit relativ einfacher Hardware wie Schieberegistern zu relativ geringen Kosten ermöglicht.

2. Wie funktioniert SPI?

SPI oder Serial Peripheral Interface wurde von Motorola in den 1980er Jahren als standardisierte, kostengünstige und zuverlässige Schnittstelle zwischen dem Mikrocontroller (anfangs Mikrocontroller von Motorola) und seinen Peripherie-ICs entwickelt.

Aufgrund seiner einfachen Schnittstelle, Flexibilität und Benutzerfreundlichkeit wurde SPI zu einem Standard und bald begannen andere Halbleiterhersteller, es in ihre Chips zu implementieren.

Im SPI-Protokoll sind die Geräte in einer Master-Slave-Beziehung in einer Mehrpunktschnittstelle verbunden. Bei dieser Art von Schnittstelle gilt ein Gerät als Master des Busses (in der Regel ein Mikrocontroller) und alle anderen Geräte (periphere ICs oder sogar andere Mikrocontroller) werden als Slaves betrachtet.

Im SPI-Protokoll kann es nur einen **Master**, aber viele **Slave-Geräte** geben.

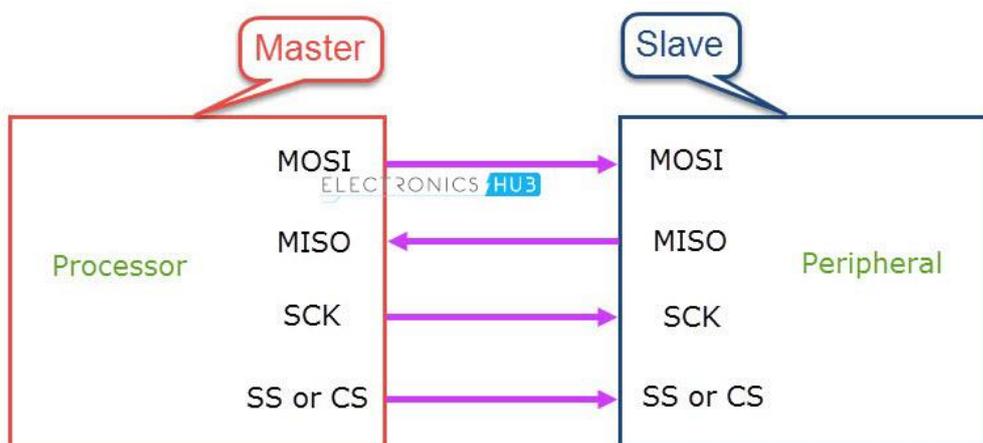
Der SPI-Bus besteht aus 4 Signalen oder Pins. Diese sind

- Master - Ausgang / Slave - Eingang (**MOSI**)
- Master - Eingang / Slave - Ausgang (**MISO**)
- Serieller Takt (**SCLK**)
- Chip Select (**CS**) oder Slave Select (**SS**)

HINWEIS: Verschiedene Hersteller verwenden unterschiedliche Nomenklaturen (Namen) oder für den SPI-Bus. Genaue Informationen entnehmen Sie bitte dem Datenblatt.

Da der SPI-Bus mit 4 Signalen oder Drähten implementiert ist, wird er manchmal auch als Four Wire Interface bezeichnet. Betrachten wir zunächst eine einfache Schnittstelle zwischen einem einzelnen Master und einem einzelnen Slave, die über das SPI-Protokoll verbunden sind, und erläutern dann die 4 Drähte.

Das folgende Bild zeigt einen Master (Prozessor), der über den SPI-Bus mit einem Slave (Peripheriegerät) verbunden ist.



Master - Out / Slave - In oder **MOSI**, wie der Name schon sagt, sind die Daten, die vom **Master** erzeugt und vom **Slave** empfangen werden. Daher sind die **MOSI**-Pins sowohl am **Master** als auch am **Slave** miteinander verbunden. Master - In / Slave - Out oder **MISO** sind die vom **Slave** erzeugten Daten, die an den **Master** übertragen werden müssen.

Die **MISO**-Pins von **Master** und **Slave** sind miteinander verbunden. Auch wenn das Signal **MISO** vom **Slave** erzeugt wird, wird die Leitung vom **Master** gesteuert. Der **Master** erzeugt ein Taktsignal an **SCLK**, das dem Takteingang des **Slaves** zugeführt wird. **Chip Select** (**CS**) oder **Slave Select** (**SS**) wird verwendet, um einen bestimmten **Slave** durch den **Master** auszuwählen.

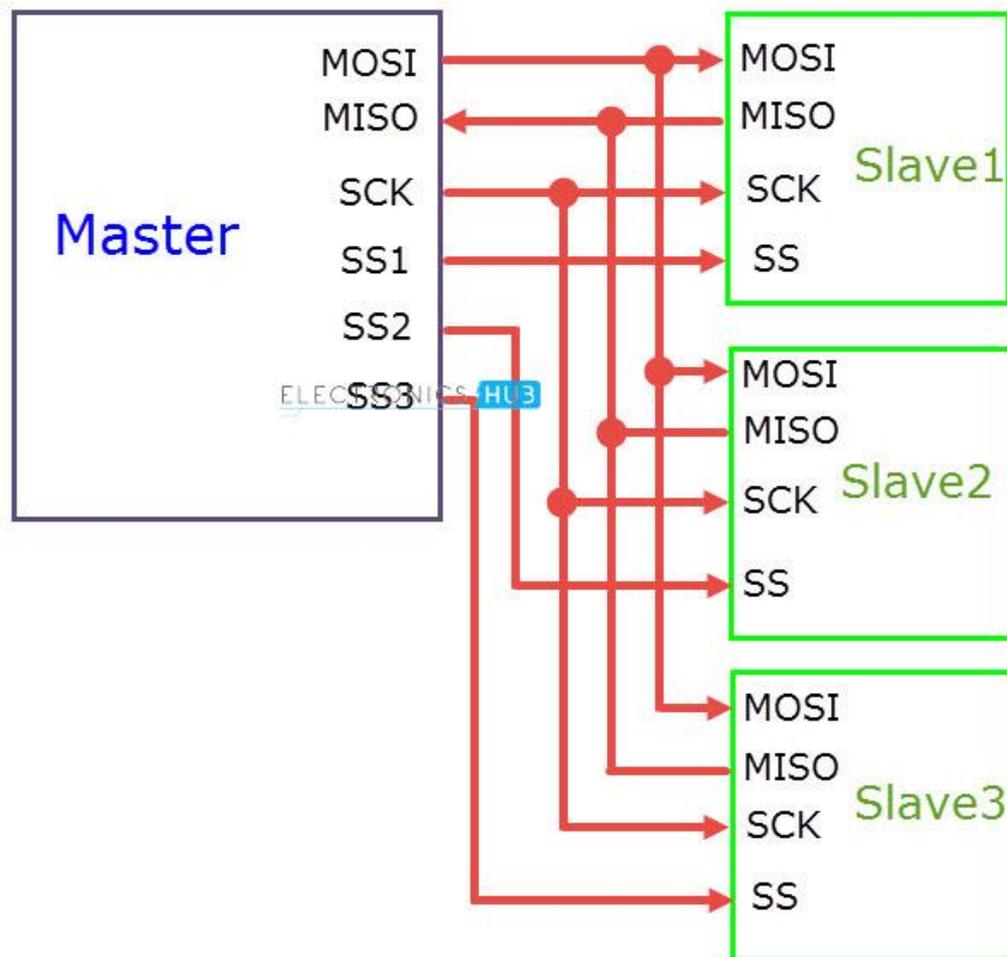
Da der Takt vom Master erzeugt wird, wird der Datenfluss vom Master gesteuert. Bei jedem Taktzyklus wird ein Datenbit vom Master zum Slave und ein Datenbit vom Slave zum Master übertragen.

Dieser Prozess findet gleichzeitig statt und nach 8 Taktzyklen wird ein Byte Daten in beide Richtungen übertragen.

Wenn die Daten nur von einem Gerät übertragen werden müssen, dann muss das andere Gerät etwas senden (sogar Müll oder Junk-Daten), und es liegt an dem Gerät, ob die übertragenen Daten tatsächlich Daten sind oder nicht.

Das bedeutet, dass für jedes Bit, das von einem Gerät übertragen wird, das andere Gerät ein Bit Daten senden muss, d.h. der Master sendet gleichzeitig Daten auf der MOSI-Leitung und empfängt Daten vom Slave auf der MISO-Leitung.

Wenn der Slave die Daten übertragen will, muss der Master das Taktsignal entsprechend erzeugen, indem er weiß, wann der Slave die Daten im Voraus senden will. Wenn mehr als ein Slave an den Master angeschlossen werden soll, sieht der Aufbau ähnlich aus wie in der folgenden Abbildung.

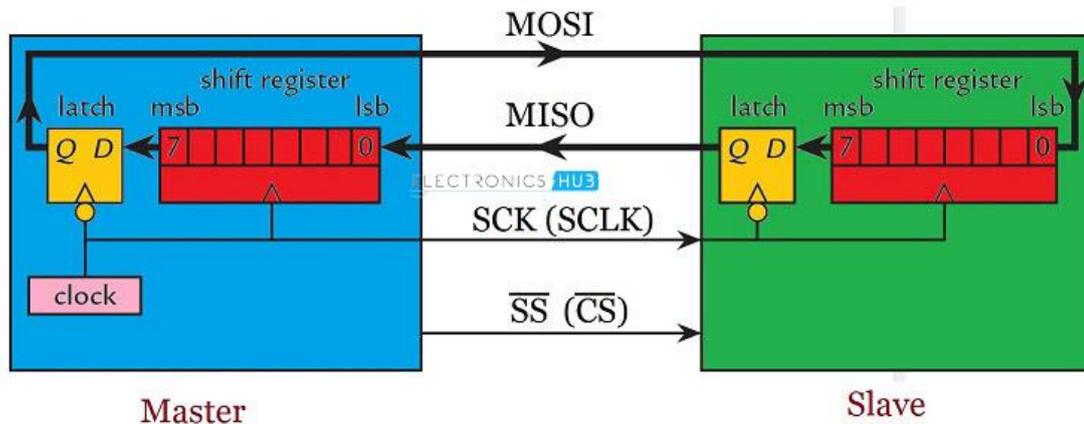


Auch wenn mehrere Slaves mit dem Master im SPI-Bus verbunden sind, ist immer nur ein Slave aktiv. Um den Slave auszuwählen, zieht der Master die SS- (Slave Select) oder CS- (Chip Select) Leitung des entsprechenden Slaves herunter.

Daher muss für jedes Slave-Gerät ein eigener CS-Pin am Master vorhanden sein. Wir müssen die SS- oder CS-Leitung herunterziehen, um den Slave auszuwählen, da diese Leitung aktiv low ist.

3. SPI-Hardware

Die Hardwareanforderungen für die Implementierung von SPI sind im Vergleich zu UART und I²C sehr einfach. Nehmen wir an, ein Master und ein einzelner Slave sind über den SPI-Bus verbunden. Die folgende Abbildung zeigt die minimalen Systemanforderungen für beide Geräte.



Wie aus dem Bild hervorgeht, besteht das Master-Gerät aus einem Schieberegister, einem Datenlatch und einem Taktgenerator. Der Slave besteht aus ähnlicher Hardware: einem Schieberegister und einem Datenlatch. Beide Schieberegister sind miteinander verbunden und bilden eine Schleife. Die Größe des Registers beträgt in der Regel 8 - Bit, aber auch größere Register mit 16 - Bit sind üblich.

Während der positiven Flanke des Taktsignals lesen beide Geräte (Master und Slave) das Eingangsbit in das LSB des Registers. Während des negativen Zyklus des Taktsignals legen sowohl der Master als auch der Slave ein Bit auf den entsprechenden Ausgang des MSB des Schieberegisters.

Somit wird bei jedem Taktzyklus ein Datenbit in jede Richtung übertragen, d. h. vom Master zum Slave und vom Slave zum Master. Für die Übertragung eines Bytes von Daten von jedem Gerät aus werden also 8 Taktzyklen benötigt.

4. SPI-Betriebsmodi

Wir haben bereits gesehen, dass es die Aufgabe des Master-Geräts ist, das Taktsignal zu erzeugen und es an den Slave zu verteilen, um die Daten zwischen Master und Slave zu synchronisieren. Die Arbeit des Masters endet nicht mit der Erzeugung eines Taktsignals mit einer bestimmten Frequenz. Vielmehr müssen sich Master und Slave auf bestimmte Synchronisationsprotokolle einigen. Hierfür kommen zwei Merkmale des Taktes ins Spiel, nämlich die Takt polarität (CPOL oder CKP) und die Taktphase (CPHA).

Die Takt polarität bestimmt den Zustand des Takts. Wenn CPOL LOW ist, ist der vom Master erzeugte Takt, d. h. SCK, im Ruhezustand LOW und wechselt im aktiven Zustand (während einer Übertragung) auf HIGH. Wenn CPOL HIGH ist, ist SCK im Ruhezustand HIGH und im aktiven Zustand LOW.

Clock Phase bestimmt den Taktübergang, d.h. ansteigend (LOW zu HIGH) oder abfallend (HIGH zu LOW), bei dem die Daten übertragen werden. Wenn CPHA gleich 0 ist, werden die Daten bei der steigenden Flanke des Taktes übertragen. Wenn CPHA 1 ist, werden die Daten bei der fallenden Flanke übertragen.

Je nach den Werten der Takt polarität (CPOL) und der Taktphase (CPHA) gibt es 4 Betriebsarten von SPI: Modi 0 bis 3.

4.1. Mode 0

Modus 0 tritt auf, wenn die Taktpolarität LOW und die Taktphase 0 ist (CPOL = 0 und CPHA = 0). Im Modus 0 erfolgt die Datenübertragung während der steigenden Flanke des Taktsignals.

4.2. Modus 1

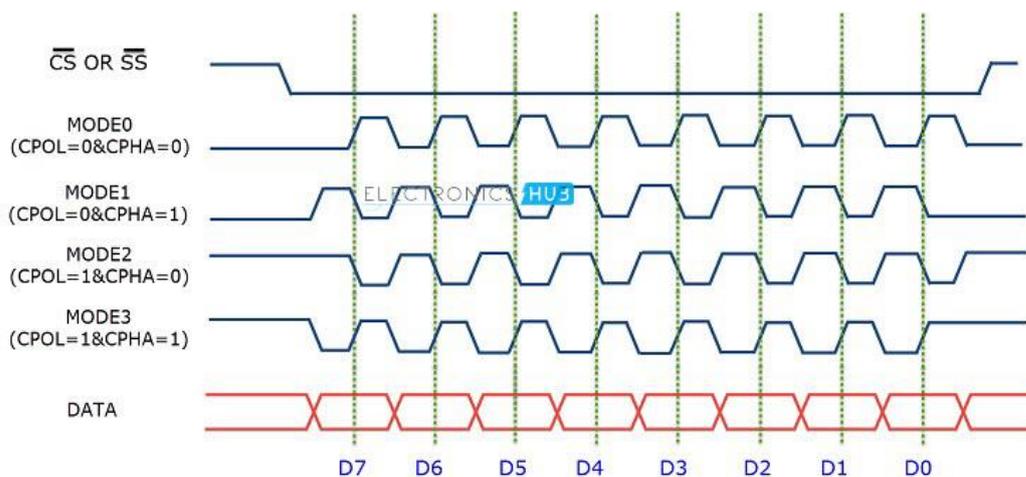
Modus 1 tritt auf, wenn die Taktpolarität LOW und die Taktphase 1 ist (CPOL = 0 und CPHA = 1). Im Modus 1 erfolgt die Datenübertragung während der fallenden Flanke des Taktsignals.

4.3. Modus 2

Modus 2 tritt auf, wenn die Taktpolarität HIGH und die Taktphase 0 ist (CPOL = 1 und CPHA = 0). In Modus 2 erfolgt die Datenübertragung während der steigenden Flanke des Taktsignals.

4.4. Modus 3

Modus 3 tritt auf, wenn die Taktpolarität HIGH und die Taktphase 1 ist (CPOL = 1 und CPHA = 1). In Modus 3 erfolgt die Datenübertragung während der steigenden Flanke des Taktsignals.

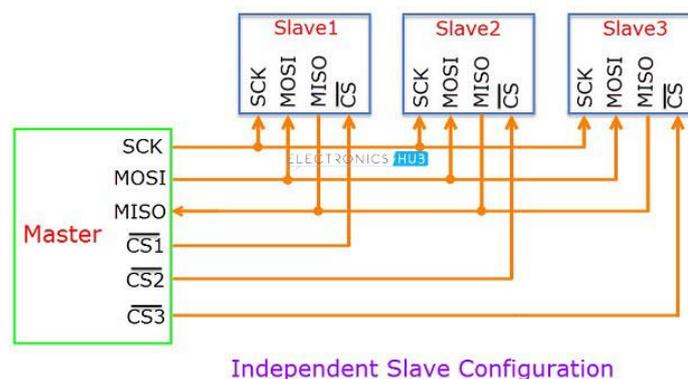


5. SPI-Konfigurationen

Es gibt zwei Arten von Konfigurationen, in denen die SPI-Geräte in einem SPI-Bus verbunden werden können. Es handelt sich um die unabhängige Slave-Konfiguration und die Daisy-Chain-Konfiguration.

In der Independent Slave Configuration hat der Master dedizierte Slave Select Lines für alle Slaves und jeder Slave kann einzeln ausgewählt werden. Alle Taktsignale der Slaves sind mit dem SCK des Masters verbunden.

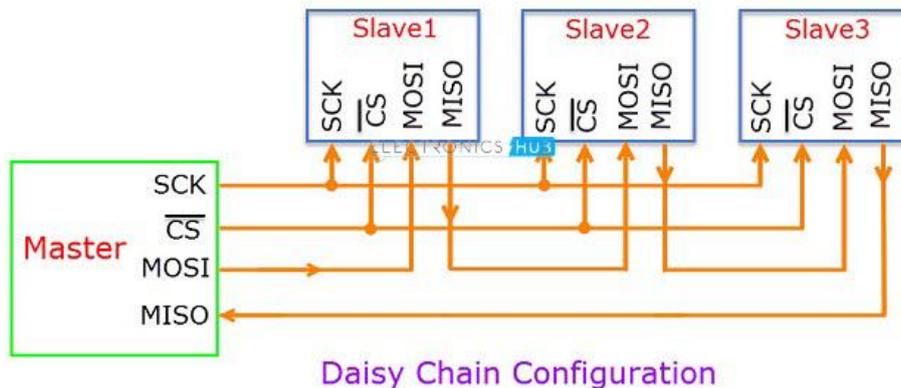
Ebenso sind alle MOSI-Pins aller Slaves mit dem MOSI-Pin des Masters und alle MISO-Pins aller Slaves mit dem MISO-Pin des Masters verbunden.



In der Daisy-Chain-Konfiguration ist nur eine einzige Slave-Select-Leitung mit allen Slaves verbunden. Der MOSI des Masters ist mit dem MOSI von Slave 1 verbunden. MISO von Slave 1 ist mit MOSI von Slave 2 verbunden und so weiter. Das MISO des letzten Slaves ist mit dem MISO des Masters verbunden.

Nehmen wir an, der Master überträgt 3 Datenbytes an den SPI-Bus. Zuerst wird das 1. Datenbyte zum Slave 1 verschoben. Wenn das 2. Byte der Daten Slave 1 erreicht, wird das erste Byte zu Slave 2 geschoben.

Wenn schließlich das dritte Datenbyte beim ersten Slave ankommt, wird das erste Datenbyte zum Slave 3 und das zweite Datenbyte zum zweiten Slave geschoben.



Wenn der Master Informationen von den Slaves abrufen will, muss er 3 Bytes Junk-Daten an die Slaves senden, damit die Informationen in den Slaves zum Master gelangen.

6. Anwendungen von SPI

- Speicher: SD-Karte, MMC, EEPROM, Flash
- Sensoren: Temperatur und Druck
- Steuergeräte: ADC, DAC, digitale POTS und Audiocodec
- Kameraobjektivhalterung, Touchscreen, LCD, RTC, Videospiel-Controller, etc.

7. Vorteile

- SPI ist sehr einfach zu implementieren und die Hardwareanforderungen sind nicht sehr komplex
- Unterstützt jederzeit eine Voll-Duplex-Kommunikation
- Sehr hohe Geschwindigkeit der Datenübertragung
- Keine Notwendigkeit für individuelle Adressen für Slaves, da CS oder SS verwendet wird
- Es wird nur ein Master-Gerät unterstützt, so dass es nicht zu Konflikten kommen kann
- Der Takt des Masters wird auf der Grundlage der Geschwindigkeit des Slaves konfiguriert, so dass sich der Slave nicht um den Takt kümmern muss

8. Nachteile

- Jeder zusätzliche Slave benötigt einen zusätzlichen dedizierten Pin am Master für CS oder SS
- Es gibt keinen Quittierungsmechanismus und somit keine Bestätigung des Datenempfangs
- Das langsamste Gerät bestimmt die Geschwindigkeit der Übertragung
- Es gibt keine offiziellen Normen und wird daher oft in anwendungsspezifischen Implementierungen verwendet
- Es gibt keine Flusskontrolle.

Einige Teile des Textes wurden zur besseren Übersicht farblich gestaltet.

Die Nutzung erfolgt auf eigenes Risiko.

Ich wünsche viel Spaß beim Bauen und programmieren

Achim

myroboter@web.de

Quellenangabe:

Der verwendete Artikel stammt aus dem englischen und wurde übersetzt.

<https://edistechlab.com/wie-funktioniert-spi/?v=3a52f3c22ed6>

<https://www.electronicshub.org/basics-serial-peripheral-interface-spi>