Workshop Mikrorechner des DARC e.V. HOB

02.06.2013, DK4AQ

Theorie SPI Bus

1. Historie und Zweck des SPI-Bus

Der Serial Peripheral Bus (SPI) wurde von Motorola ca. 1979 mit einem ihrer damals populären 8-Bit-Prozessoren eingeführt. Er verband Prozessor und Peripheriebausteine mit 4 Leitungen. Damit ist er ein Vorläufer des I2C-Bus. Es ist kaum eine formale Spezifikation des Busses zu finden, man muss schon Datenblätter und Applikationsberichte der damaligen Mikrorechner lesen.

Ähnlich wie beim I2C gibt zahlreiche Abweichungen der Implmentierungen voneinander. Da obere Ebenen des Übertragungsprotokolls nicht definiert wurden, ist die Interpretation auf der Byte-Ebene sehr unterschiedlich.

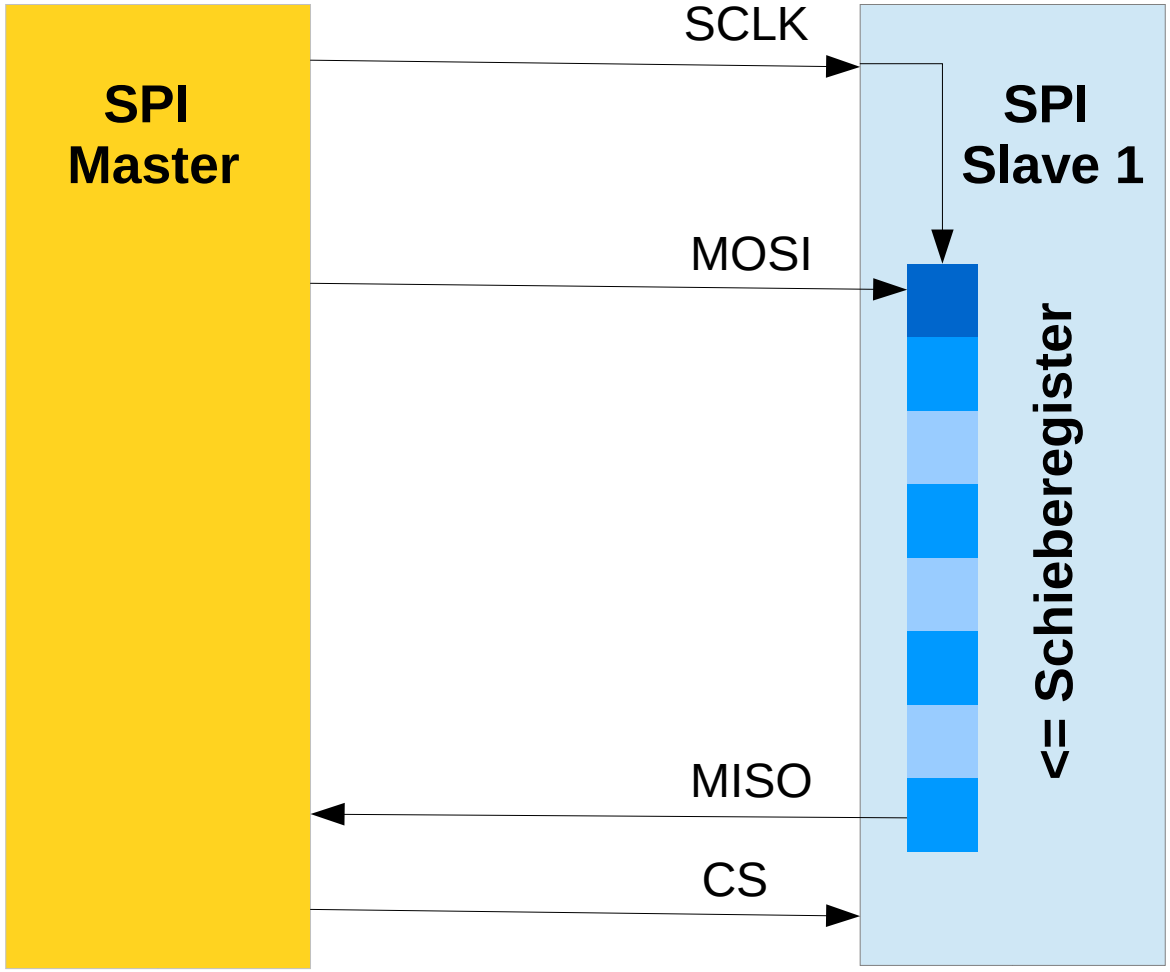
Der SPI-Bus eignet sich insbesondere für den schnellen Transport großer Datenmengen.

2. Übertragungsprinzip mit MISO, MOSI, SCLK und CS

SPI ist ein einfach gestaltetes Master-Slave-System. Die Datenübertragung erfolgt synchron.

- Das Clock-Signal SCLK wird vom Bus-Master gesendet und bestimmt, wann der Empfänger den Datenwert übernimmt Alle SPI-Signal sind synchron zu diesem Clocjk.
- Die Daten zum Slave werden vom Signal MOSI (Master Out slave In) übertragen.
- Die Daten zum Master werden vom Signal MISO (Master In Slave Out) übertragen

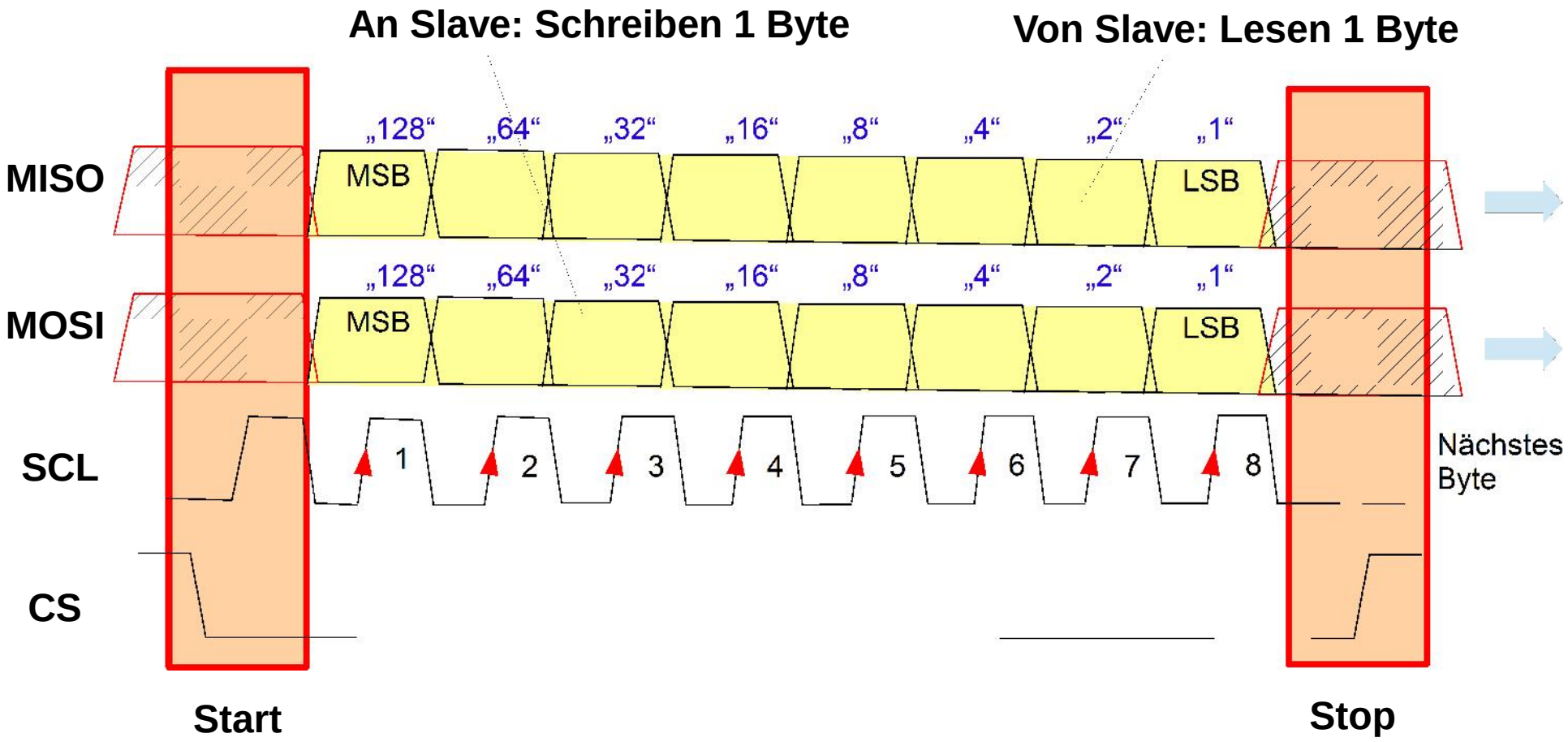
-SPI enthält keine Adressierung von Bausteinen, deshalb wird ein angesprochener Baustein über ein Chip-Select-Signal aktiviert.



Es gibt sowohl Bausteine, die diesen ringförmigen Datenverkehr unterstützen, als auch Bausteine, die nur eine Richtung unterstützen.

Auch die Bedeutung der geschriebenen und der gelesenen Daten kann bausteinspezifisch unterschiedlich definiert sein.

Das Prinzip ist ein Datenaustausch. SPI kann über MOSI Bits in den Slave schieben, gleichzeitig kann der Slave Bits an den Master liefern.



Ob dieses Feature vom Slave unterstützt wird, ist im Slave-Datenblatt zu finden. Wenn einer der beiden Leitungen nicht ausgewertet oder bedient wird, kann man natürlich auch nur eine Richtung nutzen.

Das Chip-Selekt-Signal muss bei der Übertragung mehrerer Bytes nicht nach jedem Byte auf inaktiv gesetzt werden.

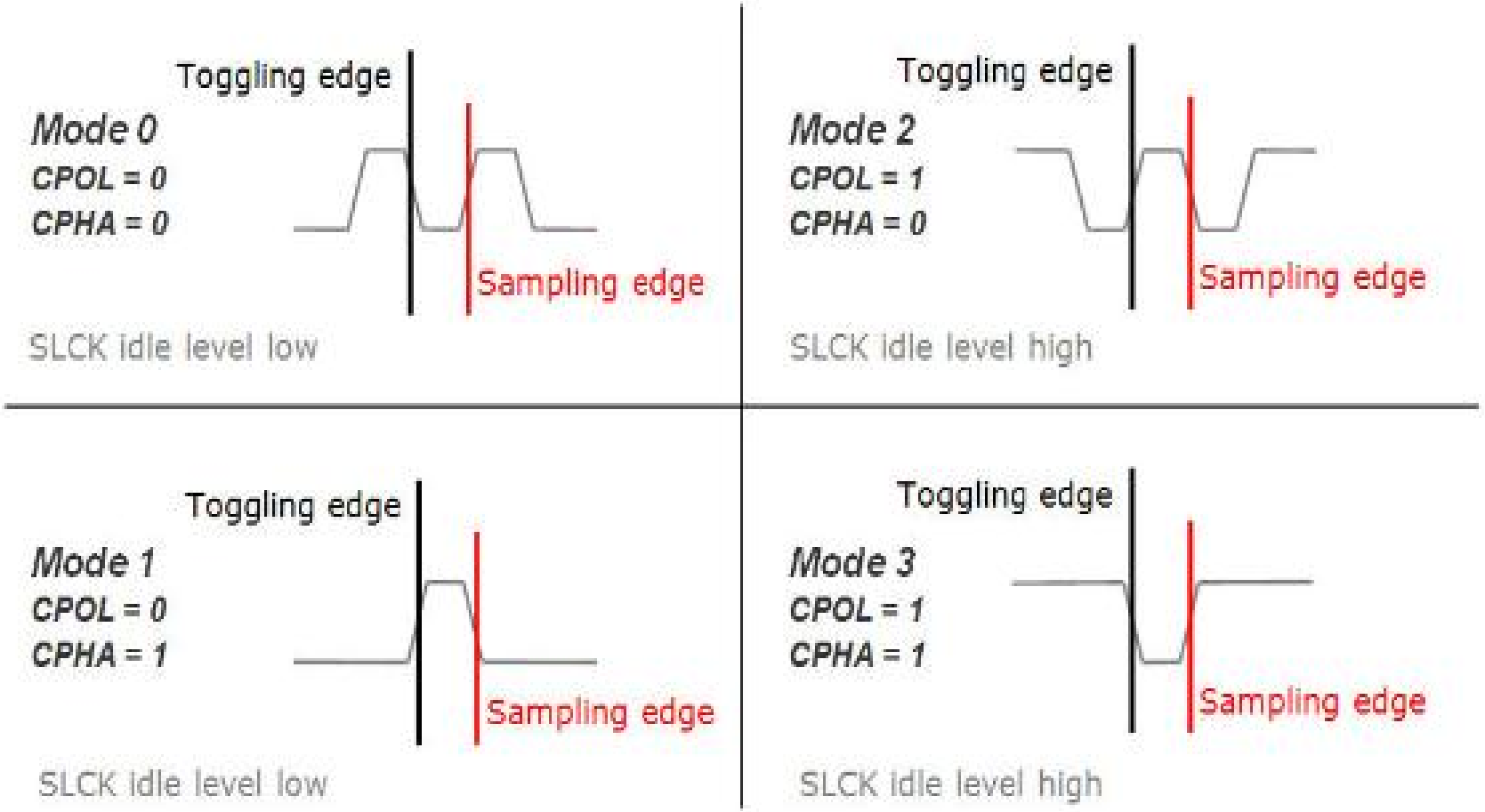
3.1 Clock-Rate

Es existieren keine standardisierten Takt-Geschwindigkeiten. Mikrocontroller lassen Einstellungen bis z.B. 12MHz zu. Heutige Peripherie-Bausteine können meist maximal 10MHz vertragen. Arduino stellt standardmässig 4MHz ein, kann jedoch auch umkonfiguriert werden.
Es gibt SPI-Adapter für PCs, die bis zu 100MHz Taktrate erlauben. Das dürfte jedoch nur mit speziellen Treberbausteinen und abgeschlossenen HF-tauglichen Leitungen funktionieren.

Übrigens: Alle Signal sind statisch, man könnte den Takt als entprelltes Signal sehr langsam handbetätigt erzeugen, auch das muss funktionieren.

3.2 Clock Betriebsarten

Die Art der Taktung kann beim SPI unterschiedlich konfiguriert werden. Es existieren 4 Modi (Mode 0,1,2,3), die im wesentlichen die SCLK-Flanke bestimmen, auf die hin die Daten übernommen werden. Die Parameter dazu sind CPOL (Clock Polarity) und CPHA (Clock Phae).



Quelle: http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/

Toggling Edge: Zu dieser Flanke sollen die Daten angelegt werden
Sampling Edge: Zu dieser Flanke werden die Daten übernommen.

3. Leitungseigenschaften

3.1 Leitungs-Pegel

Da SPI noch weniger standardisiert ist als I2C, gibt es auch keine genaue Spezifikationen für Spannungsschwellen. Daher muss der Anwender hier die Datenblätter von Master und Slave kontrollieren, ob die Schwellen passen. Da die Bauteile meist auch in unterschiedlichen Betriebsspannungen betrieben werden können, werden die Möglichkeiten hier nochmals vervielfacht. Folgende Werte sollten betrachtet werden

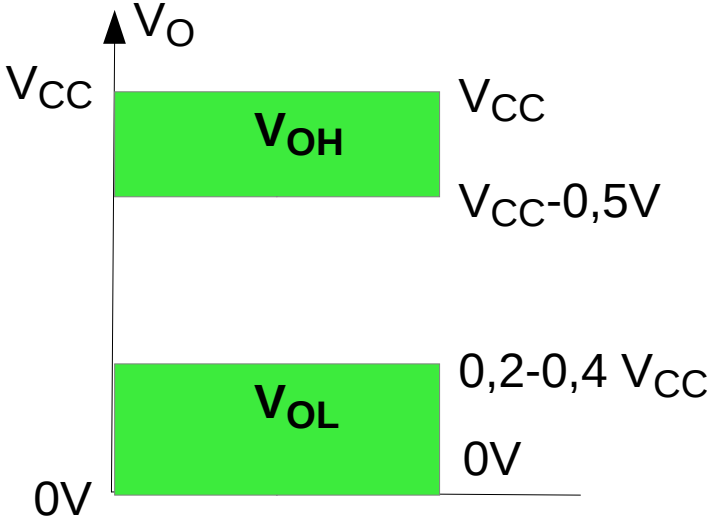
Für die Open-Collector-Variante sollten bei hohen Geschwindigkeiten die Pull-Up-Widerstände kleiner werden. Die Grenze nach unten wird durch den maximalen Ausgangsstrom vorgegeben. Dieser Wert sollte aus dem Datenblatt ermittelt werde. Die meisten Bausteine können 1-2 mA vertragen.

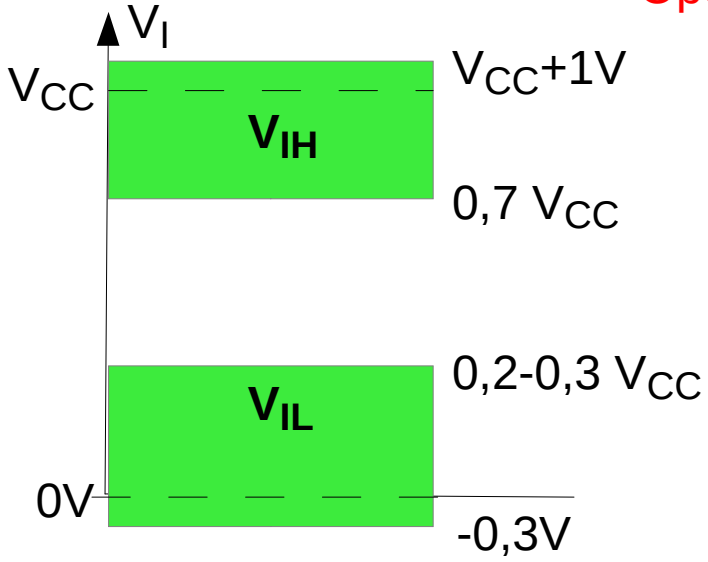
Sym.	Characteristic	Min.	Max.	Units	Test Conditions
V_{IH1}	High-level Input Voltage	0.7 V_{CC}	V_{CC} +1	V	
V_{IL1}	Low-level Input Voltage	-0.3	0.3 V_{CC}	V	$V_{CC} \geq 2.7V$ **(Note 1)**
V_{IL2}		-0.3	0.2 V_{CC}	V	$V_{CC} < 2.7V$ **(Note 1)**
V_{OL}	Low-level Output Voltage	—	0.4	V	I_{OL} = 2.1 mA
V_{OL}		—	0.2	V	I_{OL} = 1.0 mA, $V_{CC} < 2.5V$
V_{OH}	High-level Output Voltage	V_{CC} -0.5	—	V	I_{OH} = -400 µA

Beispiel: Datenausschnitt des seriellen EEPROMS 25AA020A (Microchip)

V	VCC < 2.7V (Note 1)	
V	IOL = 2.1 mA	
V	IOL = 1.0 mA, VCC < 2.5V	
V	IOH = -400 µA	

Unsymetrische Ströme für High und Low => Open Collector !





3.2 Ausgangseigenschaften

Da auf den Ausgangsleitungen nicht zurückgesendet wird, können die Daten-Ausgänge eigentlich als Gegentakt-Endstufen ausgelegt sein (TP: Totem-Pole). Es gibt aber auch die Variante in der OC (Open-Collector)-Ausgänge verwendet werden. Dies trifft häufig auf, wenn die Bausteine zwischen I2C und SPI umgeschaltet werden können. Es gibt auch Bausteine, bei denen diese Eigenschaft davon unabhängig umgeschaltet werden kann.
Hohe Geschwindigkeiten lassen sich besser mit TP-Ausgängen erreichen, da sie sowohl gegen Betriebsspannung als auch gegen Masse sehr kleine Innenwiderstände ergeben. Dadurch kann eine Leitungskapazität schnell aufladen und auch entladen werden. Diese Ausgänge brauchen eigentlich keinen Pull-Up-Widerstand gegen Betriebsspannung.

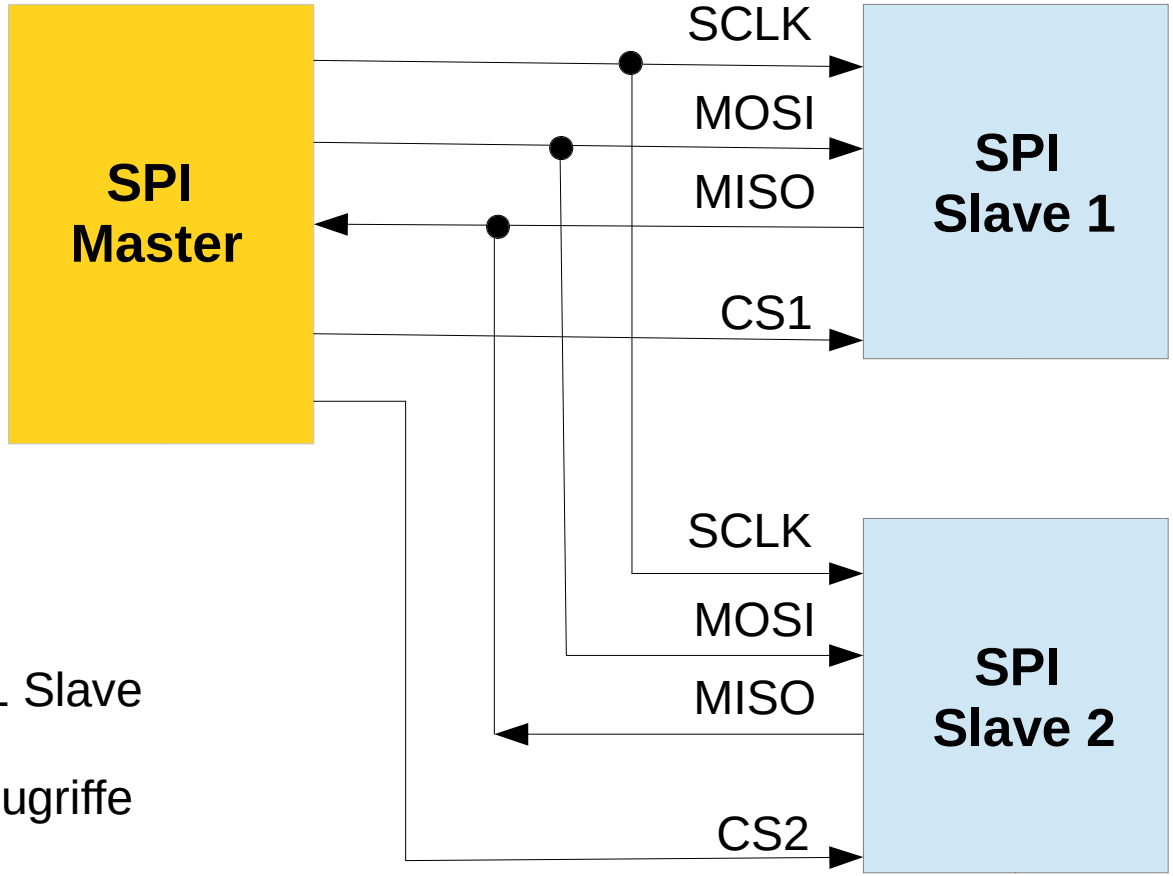
OC-Ausgängen sind nicht für hohe Geschwindigkeiten nutzbar. Zwar wird eine Leitungskapazität gegen Masse mit einem kleinen Widerstadswert entladen, jedoch durch einen grossen Pull-Up-Widerstand aufgeladen werden. Den Pull-Up-Widerstand kann man natürlich kleiner machen, jedoch begrenzt der erlaubte maximale Strom des Ausgangs diesen Wert nach unten. Ein Baustein, der 1mA Strom verkraftet, kann bei 5V Betriebsspannung also keine kleineren Pull-Ups als 4,7kOhm vertragen.

Wenn Bausteine mit ihren Ausgängen MOSI (Master) und MISO (Slave) mit Gegentakt-Ausgängen (TP) ausgestattet sind, kann es dazu kommen dass durch Chip-Selekt-Signale (CS) diese Ausgänge abgeschaltet werden (hochohmig). Dadurch kann es passieren, dass während dieser Zeit die angeschlossenen Eingänge auf einem undefinierten Potential liegen. Um dieses zu verhindern soll man auf die Leitungen ca. 10kOhm als Pull-Up-Widerstände gegen Betriebsspannung legen, auch wenn während der Datenübertragung TP-Ausgänge aktiv sind.

4. Zugriff auf Slaves

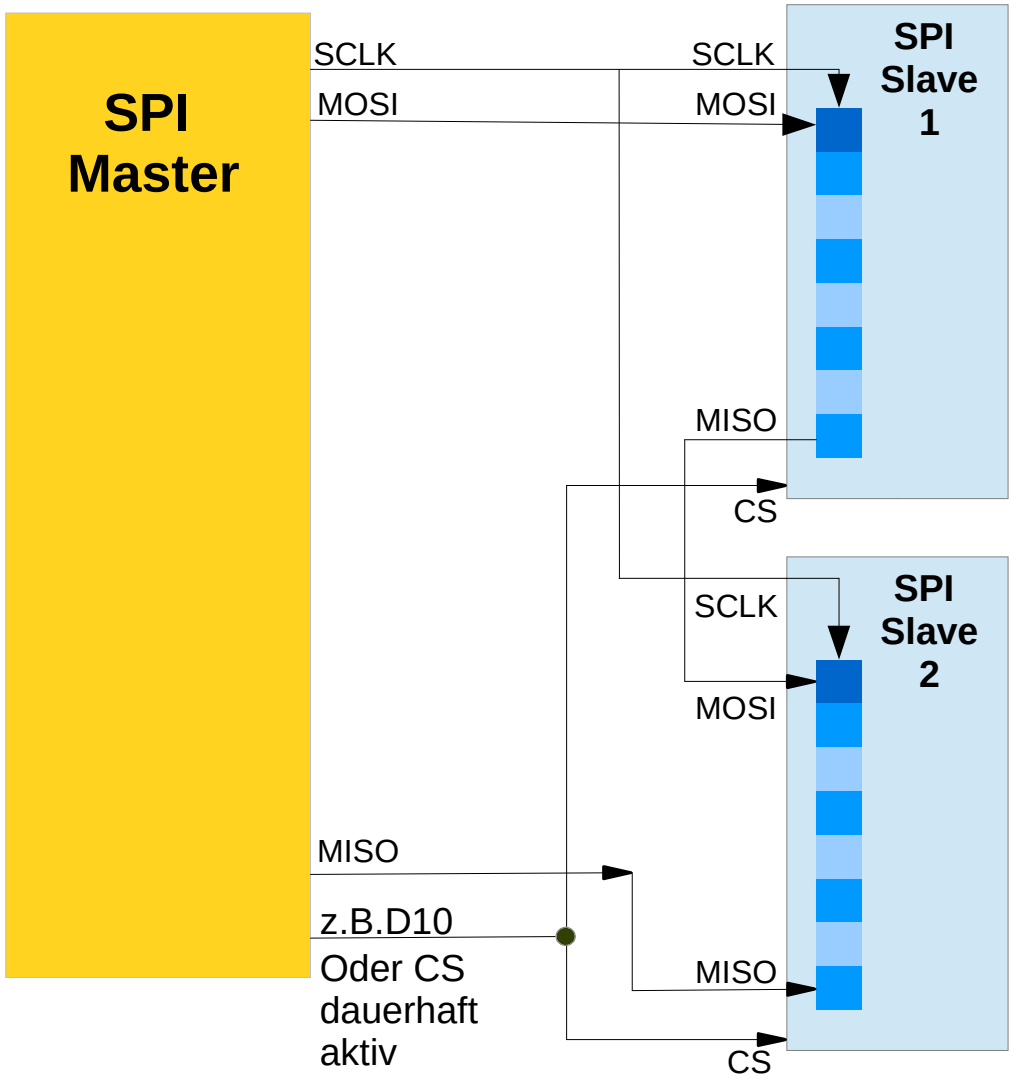
4.1 Unabhängige Selektion der Slaves

Die Slaves werden parallel mit MISO,MOSI und SCLK verbunden. Die Auswahl mit einem der Slaves geschieht durch Erzeugung eines Chip_Selekt-Signals per Portpin am Mikrorechner.



Es ist immer nur 1 Slave selektiert. Vorteil:
Schnelle gezielt Zugriffe

4.2 Seriell verkette Slaves (daisy chained)



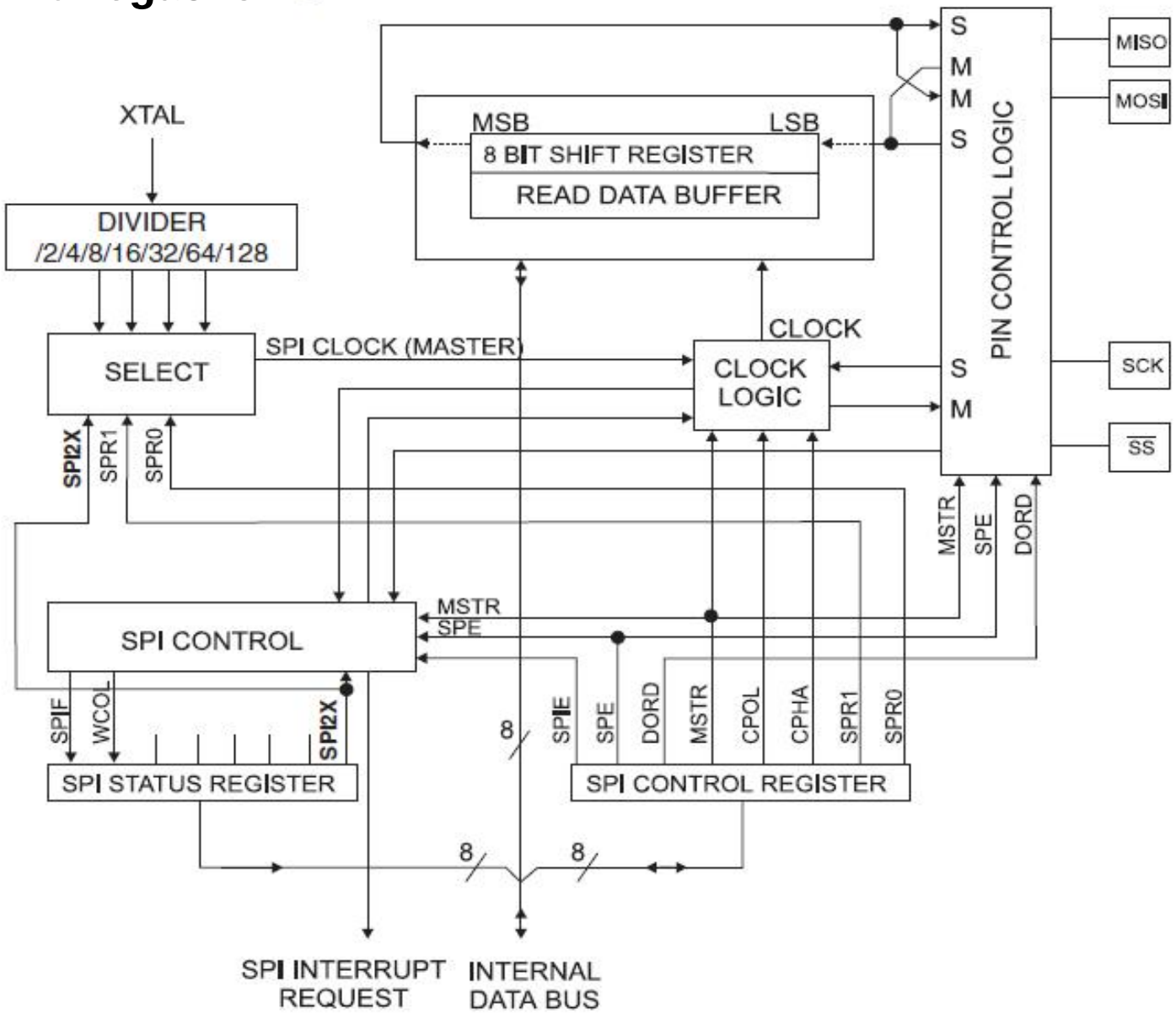
Die Slaves lassen sich in Serie betreiben. Die Ausgänge des ersten Schieberegisters geben ihre Werte an das Schieberegister des nächsten Bausteins weiter (daisy chained). Die Adressierung der Informationen an einen Baustein wird durch die Anzahl der Taktzyklen bestimmt.

Vorteil:
Einfache Verbindung

Nachteil:
Langsamerer Zugriff auf Detaildaten

5. Beispiel SPI-Hardware in Microcontrollern

5.1 Atmega328



Features

- Full-duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Seven Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wake-up from Idle Mode
- Double Speed (CK/2) Master SPI Mode

SPCR – SPI Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

SPSR – SPI Status Register

Bit	7	6	5	4	3	2	1	0	
0x2D (0x4D)	SPIF	WCOL	–	–	–	–	–	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

SPDR – SPI Data Register

Bit	7	6	5	4	3	2	1	0	
0x2E (0x4E)	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	Undefined

5.1 PIC16F87X mit Interrupt

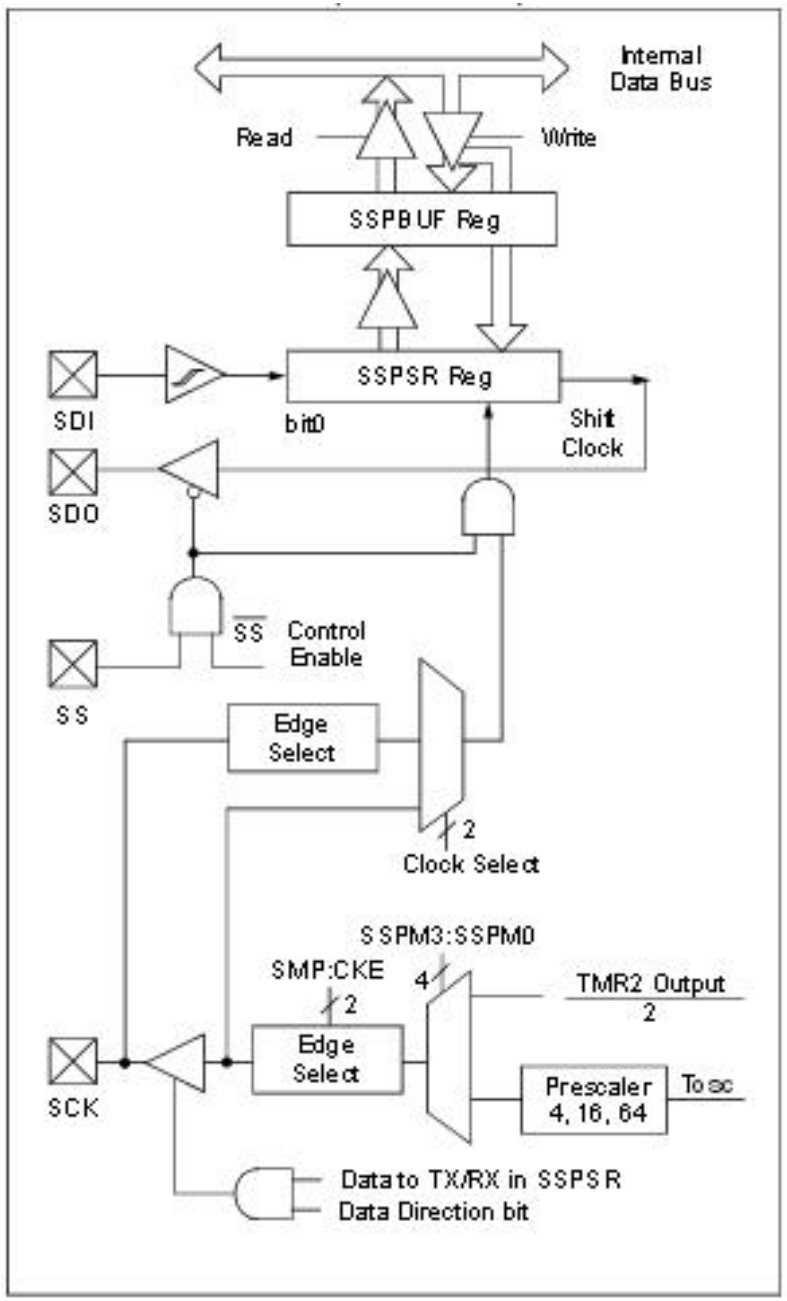


TABLE 9-1: REGISTERS ASSOCIATED WITH SPI OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR, WDT
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI mode.
Note 1: These bits are reserved on PCI16F873/876 devices; always maintain these bits clear.

Interrupt für SPI-Schnittstelle

REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 3 **SSPIF**: Synchronous Serial Port (SSP) Interrupt Flag
1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
- SPI
  - A transmission/reception has taken place.

TABLE 9-1: REGISTERS ASSOCIATED WITH SPI OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR, WDT
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI mode.
Note 1: These bits are reserved on PCI16F873/876 devices; always maintain these bits clear.

Interrupt für SPI-Schnittstelle

REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 3 **SSPIF**: Synchronous Serial Port (SSP) Interrupt Flag
1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
- SPI
  - A transmission/reception has taken place.

6. Anhänge

6.1 Vor- und Nachteile des SPI-Bus

Advantages

Quelle: (3)

- Full duplex communication
- Higher throughput than I²C or SMBus
- Complete protocol flexibility for the bits transferred
- Not limited to 8-bit words
- Arbitrary choice of message size, content, and purpose
- Extremely simple hardware interfacing
- Typically lower power requirements than I²C or SMBus due to less circuitry (including pull up resistors)
- No arbitration or associated failure modes
- Slaves use the master's clock, and don't need precision oscillators
- Slaves don't need a unique address — unlike I²C or GPIB or SCSI
- Transceivers are not needed
- Uses only four pins on IC packages, and wires in board layouts or connectors, much fewer than parallel interfaces
- At most one unique bus signal per device (chip select); all others are shared
- Signals are unidirectional allowing for easy Galvanic isolation
- Not limited to any maximum clock speed, enabling potentially high throughput

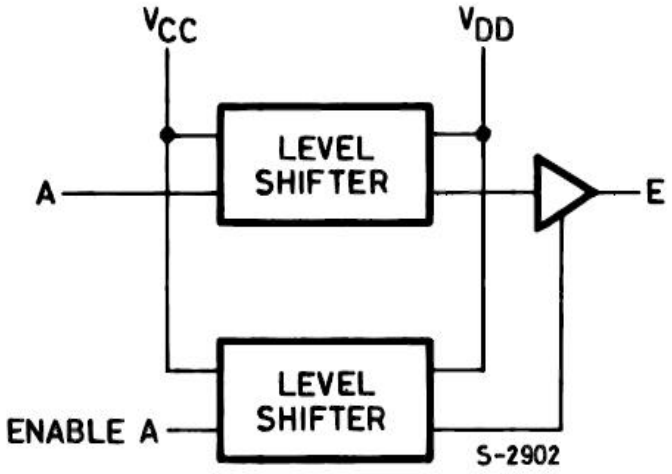
Disadvantages

- Requires more pins on IC packages than I²C, even in the three-wire variant
- No in-band addressing; out-of-band chip select signals are required on shared buses
- No hardware flow control by the slave (but the master can delay the next clock edge to slow the transfer rate)
- No hardware slave acknowledgment (the master could be transmitting to nowhere and not knowing it)
- Supports only one master device
- No error-checking protocol is defined
- Generally prone to noise spikes causing faulty communication
- Without a formal standard, validating conformance is not possible
- Only handles short distances compared to RS-232, RS-485, or CAN-bus
- Many existing variations, making it difficult to find development tools like host adapters that support those variations
- SPI does not support hot plugging (dynamically adding nodes).

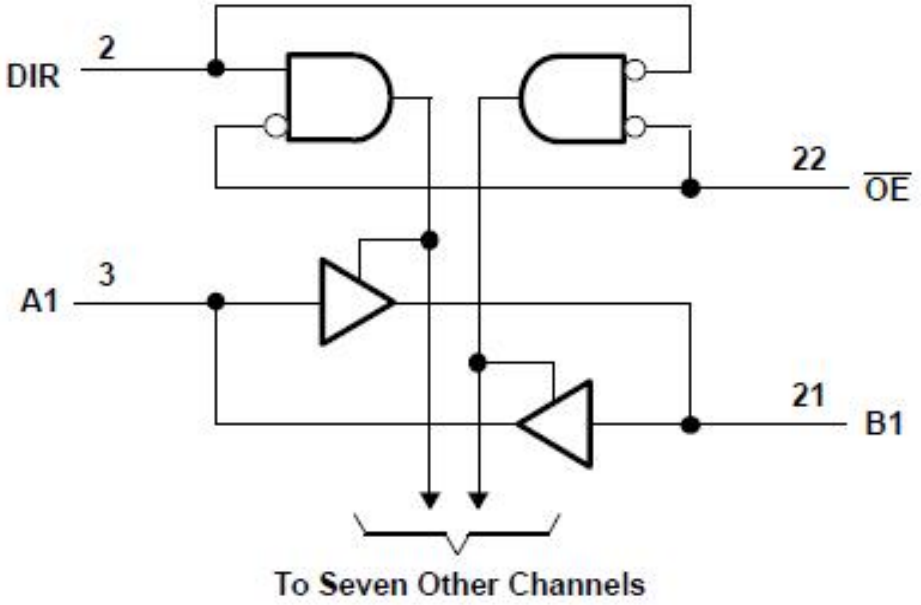
6.2 Level-Shifter

Für SPI sind bei unterschiedlichen Betriebsspannungen nur Pegelanpassungen in eine Richtung notwendig. Allerdings erfordert eine hohe Übertragungsrate etwas aufwändigere Schaltungen um eine saubere Kurvenform der Signale zu erhalten.

HCF40109 - A single 16 pin DIP that can shift 4 lines in one direction

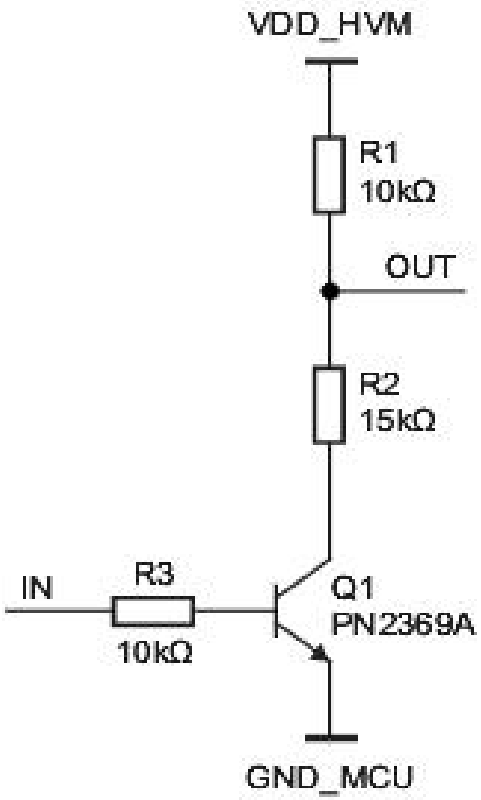


SN741VC425 -Bus transceiver and 3,3V to 5V shifter

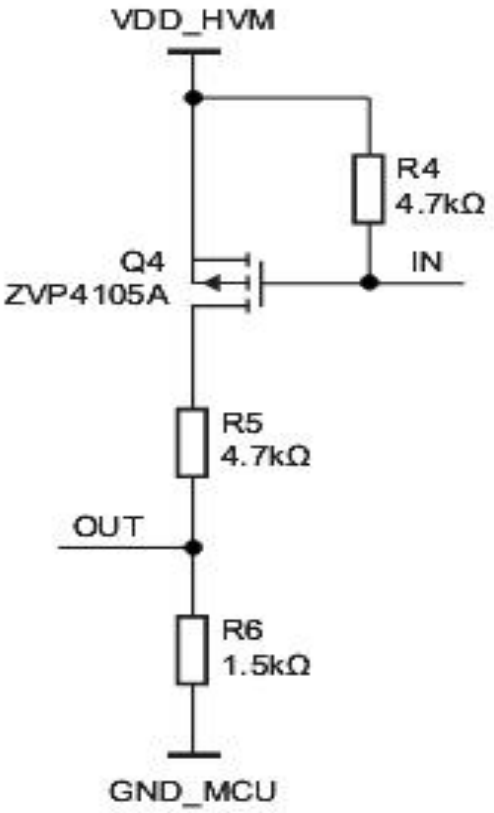


Aus (6) Atmel Aplication Note ATAN0084

Input Level-Shifter



Output Level-Shifter



6.3 Typische Fehlerursachen

Quelle: (7)

- Stimmt jeweils der SPI-Clock-Modus?
- Sind die Select-Signale vorhanden?
- Sind die erreichten Pegel ausreichend?
- Sind die Clock-Flanken sauber genug? Siehe Wellenwiderstand.
- Sind die Daten während der Datenübernahme stabil?
- Ist die Zeit zwischen Select-Signal und Übertragungsbeginn lang genug?

Quellen:

(1) Introduction to I²C and SPI protocols
http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/

(2) Arduino and the SPI bus
http://tronixstuff.wordpress.com/2011/05/13/tutorial-arduino-and-the-spi-bus/

(3) Serial Peripheral Interface Bus
http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

(4) Tutorial 8 for Arduino: SPI Interfaces
http://www.jeremyblum.com/2011/02/20/arduino-tutorial-8-spi-interfaces/

(5) Microchip PIC Programming Blog: The SPI Bus
http://microchippiclessons.blogspot.de/2009/06/spi-tutorial.html

(6) Atmel Application Note ATAN0084
http://www.atmel.com/Images/doc9272.pdf

(7) microcontroller.net, Artikel „Serial Peripheral Bus“
http://www.mikrocontroller.net/articles/Serial_Peripheral_Interface

(8) Overview and Use of the PICmicro Serial PeripheralInterface
http://ww1.microchip.com/downloads/en/devicedoc/spi.pdf