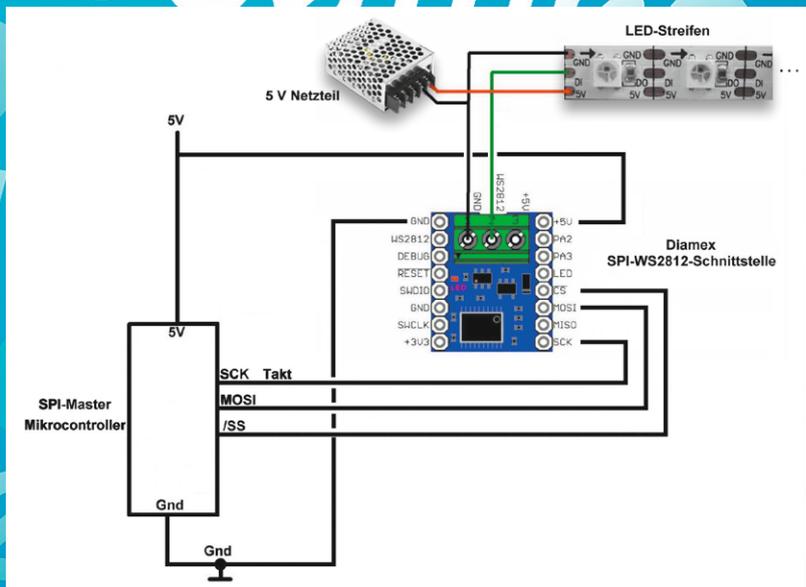
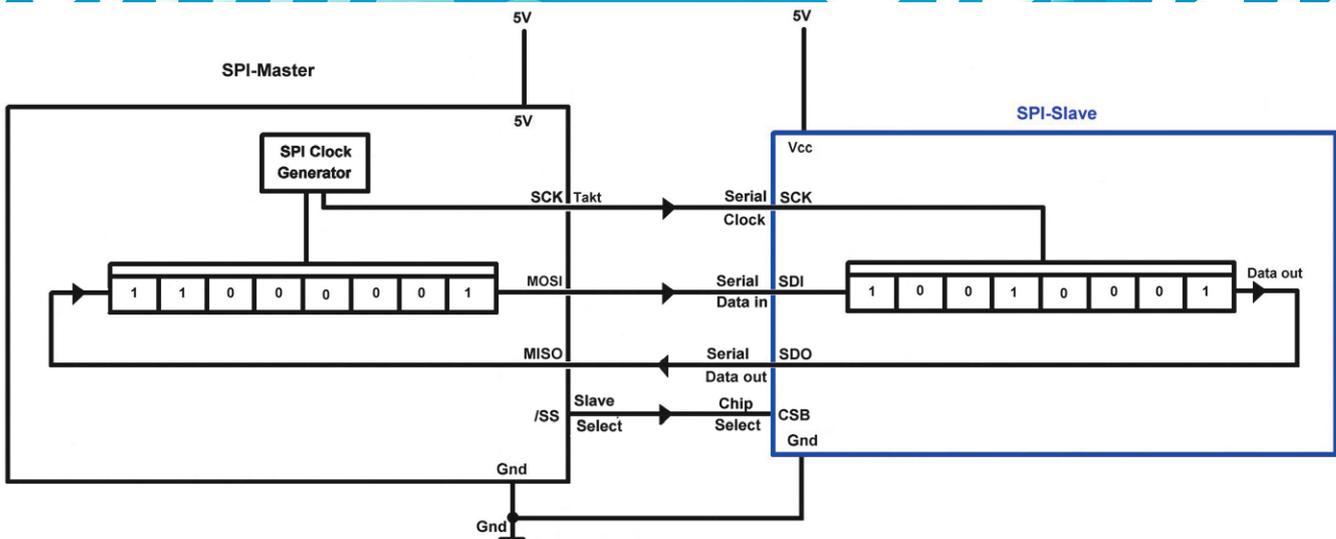




Digitale Hardwarechnittstellen

Teil 2: SPI





Die von Motorola entwickelte SPI-Schnittstelle findet im Mikrocontrollerbereich sehr oft Verwendung. SPI steht für „Serial Peripheral Interface“, also serielle Peripherie-Schnittstelle. SPI ist eine synchrone serielle Schnittstelle, was bedeutet, dass es einen Takt gibt (synchron) und die Bits hintereinander (seriell) übertragen werden. Da die SPI-Frequenz bis in den Megahertz-Bereich gehen kann und die Daten gleichzeitig in beide Richtungen übertragen werden, ist SPI für sehr schnelle Datenübertragungen geeignet. Typische Module, die mit SPI angesteuert werden, sind EEPROM-Bausteine, Digital-analog-Wandler, Analog-digital-Wandler, SD-Karten, Displays, Beschleunigungssensoren oder Echtzeituhren.

Schieberegister

Die SPI-Schnittstelle funktioniert nach dem Schieberegisterprinzip. Ein Schieberegister ist ein elektronischer Baustein, bei dem Speicherzellen in Reihe geschaltet sind. Jede Speicherzelle kann den Inhalt 1 oder 0 haben. Die Inhalte der Speicherzellen können synchron zu einem Taktsignal von einer Speicherzelle zur nächsten weitergeschoben werden.

Ein klassisches Schieberegister ist das Schieberegister 74HC595 (Best.-Nr. CL-11 37 21, CL-00 47 52). In **Bild 1** sieht man die Speicherzellen mit der Darstellung der exemplarischen Inhalte im oberen Teil des blau umrandeten Rahmens. Mit jedem Takt des an Pin 11 anliegenden Taktsignals (0-1-Wechsel = steigende Flanke) werden die Bits im Schieberegister um je eine Position nach rechts geschoben. Von links wird dabei das am Dateneingang (Pin 14) anliegende Signal in das Schieberegister hineingeschoben und der Inhalt der Speicherzelle ganz rechts wird an Pin 9 aus dem Schieberegister hinausgeschoben. Wenn am Pin 12 (Latch) ein Wechsel von 0 nach 1 (steigende Flanke) erfolgt, wird der Inhalt des Schieberegisters (oberer Teil innerhalb des blauen Kastens in **Bild 1**) in das Ausgaberegister (unten im Bild) übernommen und kann angezeigt werden. Mit Pin 13 kann die Ausgabe des Schieberegisters komplett ein- oder ausgeschal-

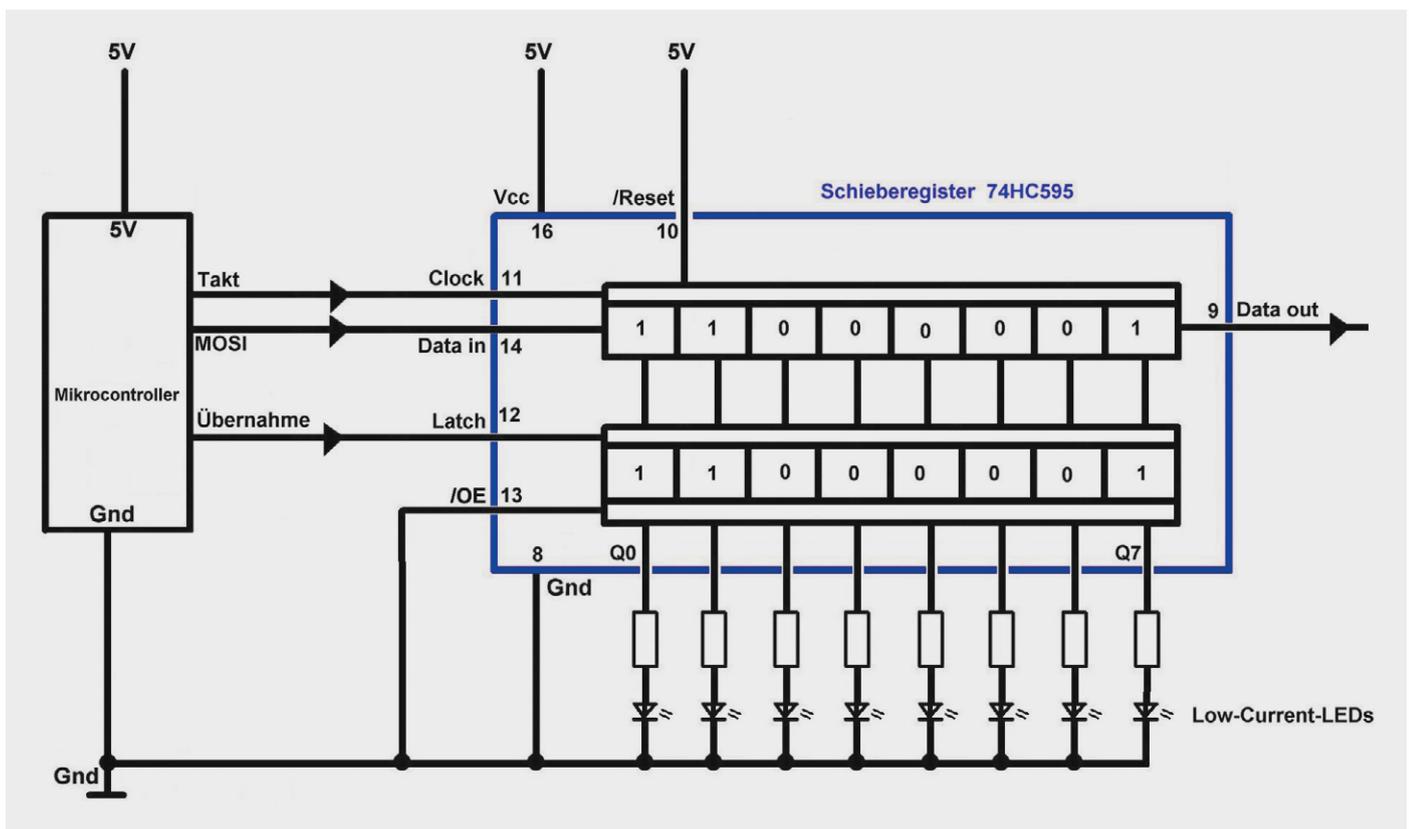


Bild 1: Schieberegister am Mikrocontroller

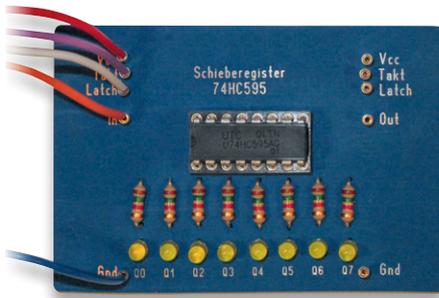


Bild 2: Signalverlauf 8-Bit-Schieberegister

tet werden (Output Enable). Das Schieberegister wird durch einen Mikrocontroller (in Bild 1 links) gesteuert. Der Mikrocontroller erzeugt das Taktsignal für das Schieberegister, gibt nacheinander die Bits (0 bzw. 1) an den Dateneingang des Schieberegisters und steuert die parallele Ausgabe der Bits. Man spricht in diesem Fall von einem SIPO-Schieberegister (Serial-in Parallel-out) im Gegensatz zu einem PISO-Schieberegister (Parallel-in Seriell-out). Es gibt Schieberegister mit unterschiedlicher Länge; Bild 1 zeigt ein 8-Bit-Schieberegister.

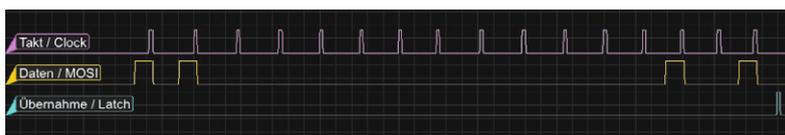
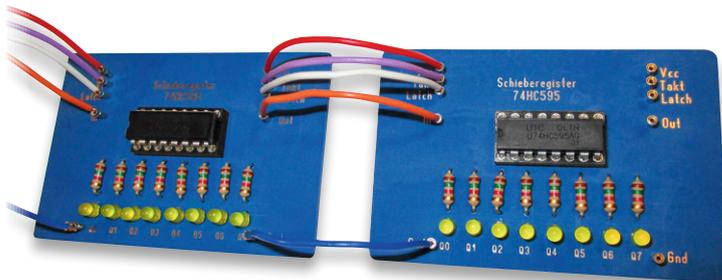


Bild 3: Signalverlauf 16-Bit-Schieberegister

Bild 2 zeigt den Signalverlauf bei der Benutzung eines Schieberegisters, wie er sich durch einen Logikanalysator (z. B. Oscium-Logikanalysator LogiScope, Best.-Nr. CL-11 53 40) darstellen lässt. Man sieht das Taktsignal (englisch: Clock), das Datensignal vom steuernden Mikrocontroller (Master) zum Schieberegister und den Verlauf des Übernahmeimpulses (englisch: Latch). Man kann auch größere, z. B. 16 Bit lange Schieberegister einsetzen oder mehrere Schieberegister hintereinanderschalten. Bild 3 zeigt den Signalverlauf bei einem 16 Bit langen Schieberegister. Es ist wiederum das Taktsignal und das anliegende Datensignal sowie – in diesem Fall nach 16 Takten – der Übernahmeimpuls zu sehen. Der große Vorteil eines Schieberegisters ist, dass man viele Bits hintereinander über wenige Leitungen (Datenleitung plus Takt und Übernahme) übertragen und durch das Schieberegister dann wieder parallel zur Verfügung stellen kann. Die Details über das 74HC595-Schieberegister sollte man sich im Datenblatt [1] ansehen.

Prinzip der SPI-Schnittstelle

Bei einer SPI-Schnittstelle gibt es einen Master, der auch den gemeinsamen SPI-Takt erzeugt, und einen oder mehrere angeschlossene SPI-Slaves (Bild 4). Im Master-Mikrocontroller gibt es ein Schieberegister, bei dem die Bits bei jedem Taktsignal nach rechts verschoben werden und das Bit ganz rechts dabei am Ausgangspin (MOSI = Master out, Slave in) zur Verfügung gestellt wird. Im Slave-Mikrocontroller gibt es ebenfalls ein Schieberegister, bei welchem die Bits im vom Master erzeugten SPI-Takt verschoben werden. Dabei werden die Bits vom Master in das Schieberegister eingeschoben und am Ende des Schieberegisters wieder hinausgeschoben. Die hinausgeschobenen Bits werden am Master am MISO- (Master in, Slave out)-Pin eingelesen. Es handelt sich also um eine Ringverbindung, bei der das Schieberegister im Master und das Schieberegister im Slave im Ring verbunden sind. Dadurch können GLEICHZEITIG (!) Bits vom Master zum Slave und vom Slave zum Master übertragen werden! Wichtig ist, dass die

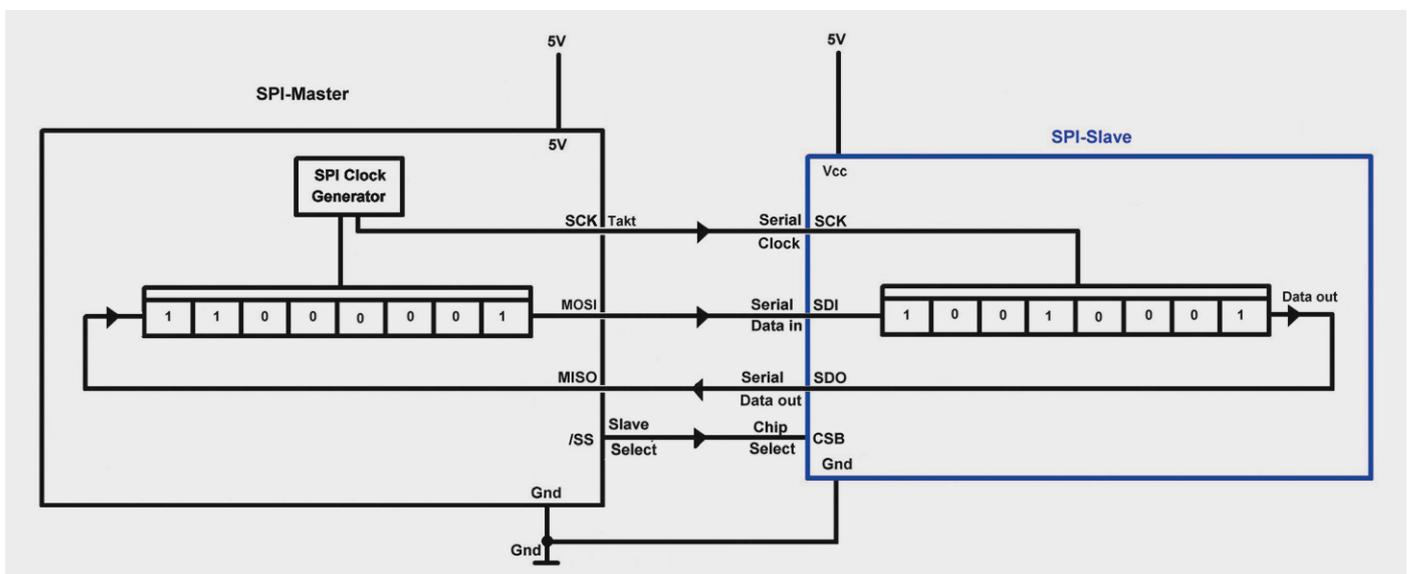


Bild 4: SPI-Prinzip

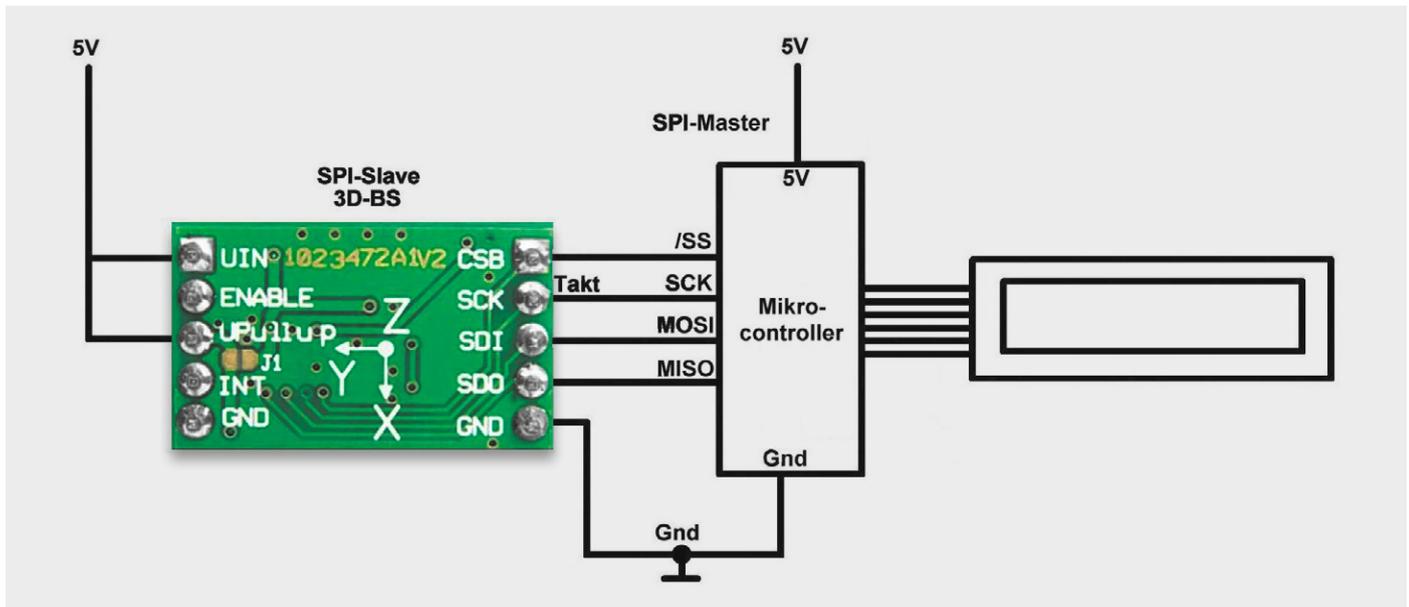


Bild 5: 3D-BS-Anschluss

Bitlängen der Schieberegister im Master und im Slave übereinstimmen. Wenn z. B. acht Bits vom Master zum Slave übertragen werden, werden gleichzeitig acht Bits vom Slave zum Master übertragen. Der Master ist normalerweise ein Mikrocontroller, der auch das Benutzerprogramm enthält, der Slave kann ein Sensor oder ein Aktor sein. Im Master-Mikrocontroller läuft ein (kompiliertes) Programm, wobei Hochsprachen oftmals SPI-Befehle zur Verfügung stellen und dem Programmierer dadurch die Arbeit abnehmen, die einzelnen Pegelwechsel manuell zu erzeugen, wodurch das Programm viel übersichtlicher bleibt.

Bei der Benutzung von SPI-Modulen muss man immer beachten, dass es verschiedene sogenannte Modi gibt, die beschreiben, mit welchen Flanken (steigend/fallend) jeweils Daten übernommen bzw. ausgegeben werden (siehe „Elektronikwissen“). Die Details muss man immer im Datenblatt des verwendeten Moduls nachlesen.

3-Achsen-Bewegungssensor

Als schönes Beispiel für ein SPI-Modul wird im Folgenden der 3-Achsen-Beschleunigungssensor 3D-BS (Bausatz: CL-09 15 21, Fertiggerät: CL-10 48 93) betrachtet. Mit diesem Modul lassen sich Beschleunigungswerte bis zu 8 g messen, und weil auch in Ruhe die Erdbeschleunigung

wirkt, kann man den Sensor auch als Lage- bzw. Neigungssensor

benutzen. Das Modul kann mit Spannungen zwischen 2,5 und 6 V betrieben werden, im Kern wird ein Beschleunigungssensor BMA020 von Bosch verwendet [2]. In Bild 5 sieht man links das Modul mit dem Beschleunigungssensor und rechts einen Mikrocontroller mit LC-Display, auf dem ein Anwendungsprogramm (z. B. in C oder BASCOM geschrieben) läuft. Man sieht in Bild 5 wieder wie oben beim Schieberegister beschrieben die Taktleitung (Takt, SCK), die Datenleitung vom Mikrocontroller zum Modul (MOSI), die Datenleitung vom Modul zum Mikrocontroller (MISO) und eine „Slave-Select“-Verbindung. Sowohl im Mikrocontroller als auch im Beschleunigungssensor auf dem Modul sind Schieberegister, die über die MOSI- und MISO-Leitungen im Ring verbunden sind.

ELV
3D-BS

x: 1.0e y: -0.4e
z: 0.1e

SPI-Modi

Motorola hat für die SPI-Kommunikation kein explizites Protokoll festgelegt, aber es hat sich die Unterscheidung von vier Modi (0, 1, 2, 3) durchgesetzt. Die Modi geben an, auf welchem logischen Level das Taktsignal im Ruhezustand sein soll und bei welchen Signalflanken Daten übernommen bzw. ausgegeben werden.

Der Ruhezustand des Taktsignals wird mit CPOL (= Clock-Polarität; Polarity) gekennzeichnet. CPOL = 0 bedeutet, dass das Taktsignal im Ruhezustand 0 ist. CPOL = 1 bedeutet einen Ruhezustand des Taktsignals mit Level 1.

Mit CPHA (= Clock-Phase; Phase) wird angegeben, ob die Daten bei der ersten Flanke

(CPHA = 0) oder bei der zweiten Flanke (CPHA = 1) übernommen werden. Bei der jeweils anderen Flanke werden die Daten an den Bus ausgegeben. Die Werte sind dem Datenblatt der jeweiligen SPI-Komponente zu entnehmen und im Programm entsprechend zu berücksichtigen.

Mode	CPOL (Takt im Ruhezustand)	CPHA (Datenübernahme bei ...)	Datenübernahme bei ...	Datenausgabe bei ...
0	0	0 ... erster Flanke	... steigender Flanke	... fallender Flanke
1	0	1 ... zweiter Flanke	... fallender Flanke	... steigender Flanke
2	1	0 ... erster Flanke	... fallender Flanke	... steigender Flanke
3	1	1 ... zweiter Flanke	... steigender Flanke	... fallender Flanke

Steigende Flanke: Wechsel des Signals von 0 nach 1

Fallende Flanke: Wechsel des Signals von 1 nach 0



1 sein. Zum Lesen ab Register 02h wird deshalb als erstes Byte 82h übertragen (80h + 02h). Unten in Bild 9 kann man die schematische Darstellung sehen.

In Bild 10 sieht man den Signalverlauf des SPI-Signals, der zum Auslesen der Beschleunigungswerte benutzt wird. Auf der MOSI-Leitung, also zum Sensor, wird eine 82h übertragen, die dem Sensor mitteilt, dass ab Register 02h gelesen werden soll. Danach werden zum Slave nur noch Nullen übertragen, aber gleichzeitig erfolgt die Übertragung der sechs Registerwerte vom Sensor zum Master (MISO). In der dekodierten Darstellung der MOSI- und MISO-Signale in Bild 10 ist sehr gut zu sehen, dass gleichzeitig sieben Bytes vom Master zum Sensor übertragen werden, wovon das erste Byte dem sogenannten Control-Byte entspricht, und dass in die andere Richtung – vom Sensor zum Mikrocontroller – ein „Dummy-Byte“ (FFh) und dann die sechs Registerwerte übertragen werden. Die Werte aus den Sensorregistern können im Mikrocontroller weiterverarbeitet und angezeigt werden.

SPI-WS2812-Schnittstelle

Ein sehr nützliches Modul von ELV ist die SPI-Schnittstelle für WS2812-LEDs (CL-12 28 55). Mit diesem Modul ist es auf sehr einfache Weise möglich, die sehr vielseitigen und innovativen RGB-LED-Streifen mithilfe weniger SPI-Befehle zu steuern. Dadurch lassen sich Anzeigen, Uhren, Beleuchtungen usw. mit den LED-Streifen oder -Ringen individuell auf SPI-Basis realisieren. Durch das preiswerte und kleine Modul

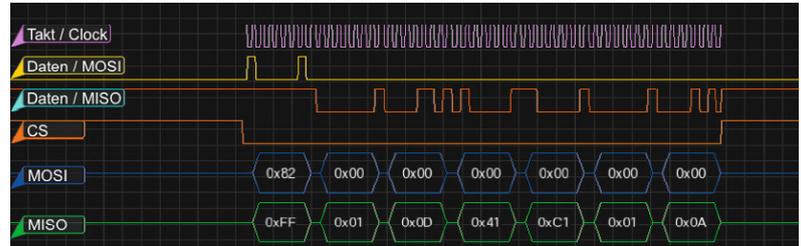


Bild 10: Signalverlauf 3D-BS Sensorwerte auslesen

kann man sich im Mikrocontroller-Programm auf die eigentliche Aufgabe konzentrieren und muss nicht die sehr zeitkritischen Routinen zur Ansteuerung der LED-Streifen im Mikrocontroller-Programm schreiben. Bild 11 zeigt den Aufbau: Ein Mikrocontroller sendet über drei Datenleitungen SPI-Befehle zu dem SPI-WS2812-Modul, welches seinerseits die Ansteuerung jeder einzelnen RGB-LED des LED-Streifens oder des LED-Rings ermöglicht. Da durch die oft hohe Anzahl von LEDs ein hoher Strombedarf entsteht, muss man den LED-Streifen mit einem eigenen, starken Netzteil mit Spannung versorgen. (Überschlagsrechnung: je RGB-LED 3x 20 mA, also 60 mA. Diese 60 mA multipliziert mit der Anzahl der RGB-LEDs. Beispiel: 60 RGB-LEDs benötigen ein Netzteil, das mindestens $60 \times 60 \text{ mA} = 3600 \text{ mA} = 3,6 \text{ A}$ liefern kann.)

Da nur Daten vom Mikrocontroller oder PC zum Modul geschrieben werden müssen, benötigt man nur die Signalleitungen SCK (Takt), /SS bzw. CS (Chip Select) und MOSI (vom Master zum Slave). MISO wird nicht benötigt. Außerdem wird eine gemeinsame Gnd-Leitung benötigt, und sowohl das steuernde System als auch das Modul und der LED-Streifen werden mit Betriebsspannung versorgt.

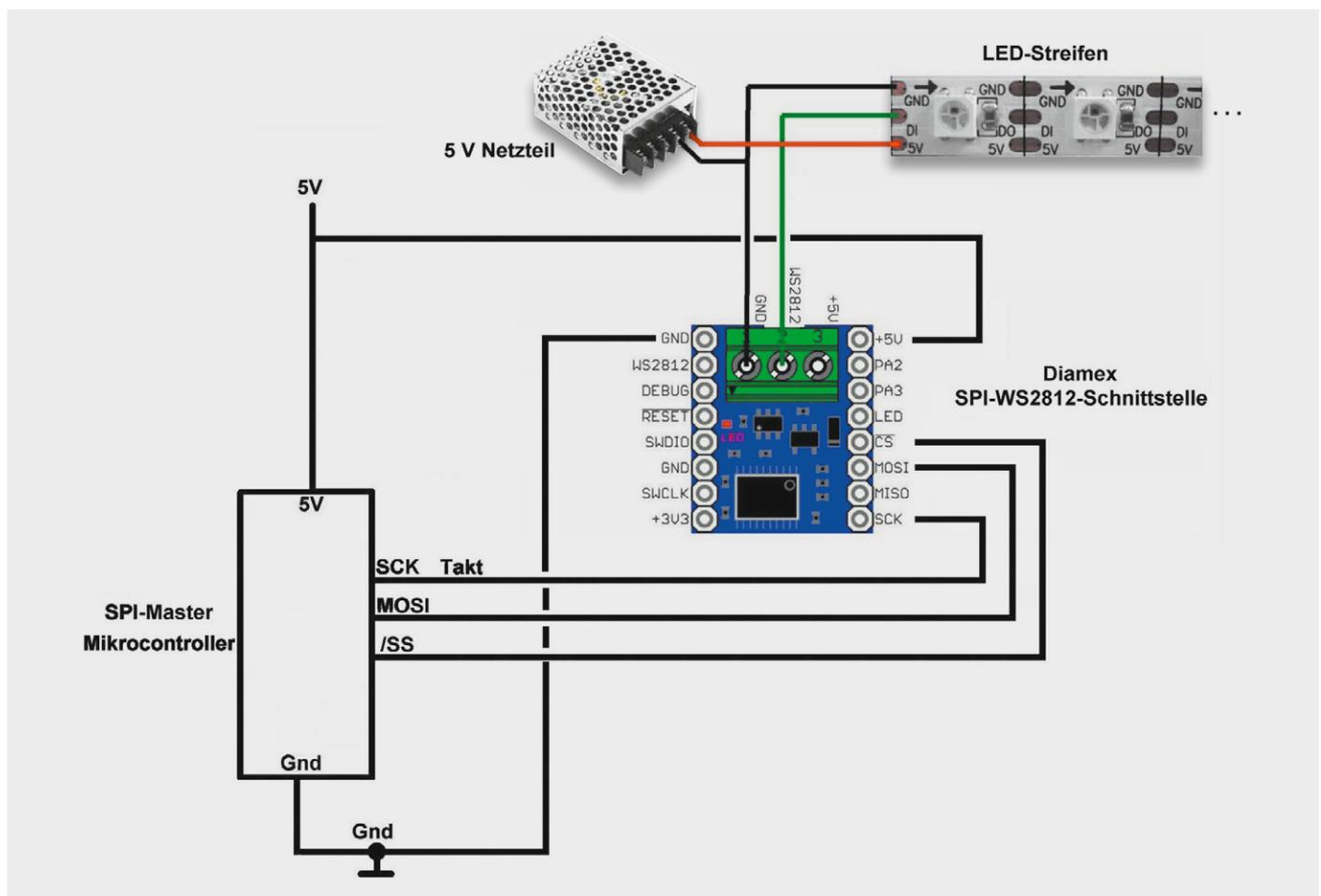


Bild 11: Anschluss SPI-WS2812-Modul



CS-LO	Befehl 1	Befehl 2	Befehl 3	Befehl 4	CS-HI
-------	----------	----------	----------	----------	-------

Bild 12: Aufbau Datenpaket SPI-WS2812-Modul (Datenblattauszug)

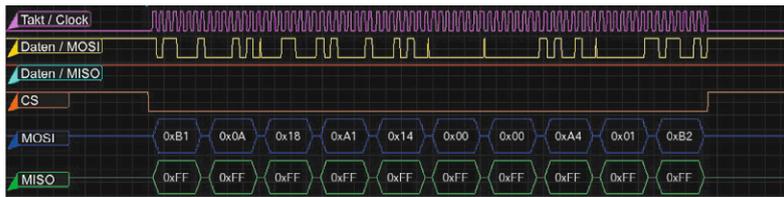


Bild 13: Signalverlauf Initialisieren und 1 rote LED

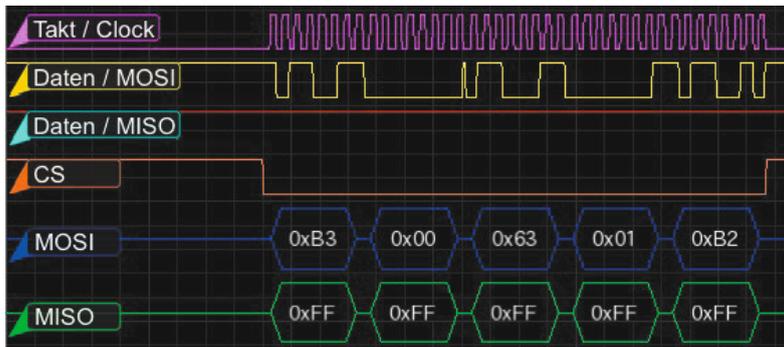


Bild 14: Signalverlauf Verschieben

Laut Datenblatt [3] wird der SPI-Mode 0 benutzt (CPOL = 0 und CPHA = 0) und die CS-Leitung ist low-aktiv, was man im Datenblattauszug in Bild 12 sieht. Im Ruhezustand ist CS also high (= 1) und wird zum Start der SPI-Übertragung auf low (0 V) geschaltet. Dann folgen (bis zu 256) Befehle und die CS-Leitung geht wieder auf high (5 V). Schematisch sieht man das in Bild 12. Im realen Signalverlauf ist das in der Logikanalysator-Darstellung in Bild 13 und Bild 14 zu sehen.

Die Ansteuerungsbefehle für das Modul stehen im Datenblatt [3]. Ein Auszug aus der Befehlsübersicht wird in Tabelle 1 gezeigt. Zum Initialisieren des Moduls wird dem Modul die Anzahl der verwendeten LEDs und die Type des LED-Streifens mit dem Befehlsbyte B1h (gefolgt von der Anzahl und der LED-Art) mitgeteilt. Eine SPI-Übertragung sieht z. B. so aus:

B1h 0Ah 18h (Anzahl LEDs: 10, LED-Controller-Typ: WS2812)
 A1h 14h 00h 00h (Farbregister setzen mit Rotwert, Grünwert, Blauwert)
 A4h 01h (Farbe aus Farbregister in LED-Register übertragen)
 B2h (alles anzeigen)

Zur Verdeutlichung wurden hier verschiedene Befehle zeilenweise dargestellt. In Bild 13 sieht man auf der MOSI-Leitung genau diesen Ablauf im Rahmen einer SPI-Übertragung. Das Modul überträgt keine Daten zurück zum Master. Man muss die MISO-Leitung nicht verkabeln. In Bild 13 sieht man für MISO nur FFh-Bytes.



Ein zweites Beispiel mit dem Modul soll weitere Sicherheit im Umgang mit der SPI-Schnittstelle bringen. Laut Tabelle 1



kann man mit dem Befehlsbyte B3h LED-Inhalte auf dem LED-Streifen verschieben. Dadurch lassen sich Lauflichter u. Ä. erzeugen. Befehlsablauf:

B3h 00h 63h 01h (ab LED 0 bis LED 63h = 99 um eine Position nach rechts verschieben)
 B2h (alles anzeigen)

In der Logikanalysator-Darstellung in Bild 14 sieht man, dass auf der MOSI-Leitung die Bytes B3h 00h 63h 01h B2 übertragen werden.

Ebenso kann man auch die ELV-DCF77-Echtzeituhr (CL-13 05 41) mit SPI ansteuern. Man kann mithilfe dieses Moduls und einiger SPI-Befehle die atomgenaue DCF77-Uhrzeit für eigene Zwecke (Uhrenprojekt, Logdateien etc.) verwenden. Wie in diesem Artikel gezeigt, muss man sich die Produktbeschreibung bzw. das Datenblatt bezüglich der Register und des SPI-Protokolls genau ansehen und dann die entsprechenden SPI-Befehle im Mikrocontroller-Programm schreiben. Dadurch hat man Schreib- oder Lesezugriff auf Aktoren oder Sensoren, die SPI unterstützen.

Fazit

Die SPI-Schnittstelle ist eine verbreitete und sehr schnelle serielle Schnittstelle, die zur Ansteuerung von Modulen und zum Auslesen von Sensoren Verwendung findet. Die Schnittstelle funktioniert nach dem Schieberegister-Prinzip, bei dem die Bits mit einem vom Master erzeugten Takt aus dem Master hinaus in einen oder mehrere Slaves und von dort wiederum in den Master geschoben werden.

Im nächsten ELV Journal wird die I²C-Schnittstelle vorgestellt, bei der angeschlossene Geräte über individuelle Adressen angesprochen werden. **ELV**



Weitere Infos:

- [1] Datenblatt Schieberegister 74HC595
www.elv.de: Webcode #10045
- [2] Datenblatt BMA020-Bewegungssensor
www.elv.de: Webcode #10046
- [3] Datenblatt DIGI-DOT-BOOSTER
www.elv.de: Webcode #10047



Befehl	Parameter 1	Parameter 2	Parameter 3	Funktion
xB1	LED_Anzahl 2...256 z. B. x0A = 10	LED-Art 24: WS2812 z. B. x18 = 24		LED-Streifen initialisieren
xA1	Rotwert 0...255 z. B. x14 = 20	Grünwert 0...255 z. B. x00	Blauwert 0...255 z. B. x00	Farbregister setzen
xA4	LED-Nummer 0 bis (LED_Anzahl - 1) z. B. x01 = LED 1			LED bekommt Farbe aus Farbregister
xB2				alles anzeigen
xB3	von_LED 0 bis (LED_Anzahl - 1) z. B. x00 = LED 0	bis_LED von_LED bis (LED_Anzahl - 1) z. B. x063 = LED 99	Anzahl_LEDs 1 bis (LED_Anzahl - 1) z. B. x01 = um 1 verschieben	verschieben

Tabelle 1: Befehlssatz SPI-WS2812-Modul (Datenblattauszug)

Preisstellung Dezember 2016 – aktuelle Preise im Web-Shop

Empfohlene Produkte	Best.-Nr.	Preis
Arduino Uno	CL-10 29 70	€ 27,95
ELV 6-Achsen-Bewegungssensor 6D-BS mit SPI- und I ² C-Schnittstelle, Komplettbausatz	CL-13 05 98	€ 15,60
3-Achsen-Beschleunigungssensor 3D-BS, Bausatz	CL-09 15 21	€ 6,95
3-Achsen-Beschleunigungssensor 3D-BS, Fertiggerät	CL-10 48 93	€ 9,95
Diamex Digi-Dot-Booster mit SPI-Schnittstelle für WS2812-LEDs	CL-12 28 55	€ 9,90
Real-Time-Clock-DCF-Modul mit I ² C-, SPI- und UART-Schnittstelle RTC-DCF, Komplettbausatz	CL-13 05 41	€ 12,95
ELV USB-SPI-Interface USB-SPI, Komplettbausatz	CL-13 12 92	€ 29,95
Diamex 4-m-RGB-LED-Streifen mit 240 WS2812-RGB-LEDs, hochflexibel	CL-11 77 43	€ 59,95
Diamex 5-m-RGB-LED-Streifen mit 150 WS2812-RGB-LEDs, hochflexibel	CL-11 77 44	€ 44,95
Diamex 4-m-RGB-LED-Streifen mit 240 WS2812-RGB-LEDs, schwarze Platine	CL-12 56 59	€ 59,95
Diamex 5-m-RGB-LED-Streifen mit 150 WS2812-RGB-LEDs, schwarze Platine	CL-12 56 60	€ 44,95
Diamex 2-m-LED-Streifen mit WS2812-LEDs, 288 LEDs, weiße Platine	CL-12 56 61	€ 49,90
60 mA pro RGB-LED rechnen:		
Netzteil Mean Well LPV-60-5, 5 Vdc, 8 A	CL-12 08 66	€ 22,95
Netzteil Mean Well LPV-100-5, 5 Vdc, 12 A, 60 W	CL-12 31 78	€ 34,95
2-Kanal-USB-Speicher-Oszilloskop LabNation Smartscope	CL-12 37 24	€ 229,95
USB-2.0-OTG-Kabel Micro-B/Mini-B für Android-Geräte, 0,5 m	CL-12 70 16	€ 5,95
Logikanalysator Oscium LogiScope	CL-11 53 40	€ 159,95
ELV Triggergenerator TG1 für SPI/I ² C/UART, Komplettbausatz	CL-14 21 24	€ 29,95
Real-Time-Clock Micro Crystal RV-2123-C2-TA-20ppm, SPI-Bus	CL-11 26 02	€ 3,25
Chip-on-Glass-LC-Display EA GOGM162S-A	CL-06 93 11	€ 9,45
Diamex Controller DD-Starter für WS2812-LEDs	CL-12 56 55	€ 9,90
Mini-Panel 8x8 mit WS2812-LEDs	CL-12 56 56	€ 24,90
Barthelme 5-m-Anschlussleitung für LED-Streifen, 5-adrig, weiß	CL-12 49 25	€ 12,95
Barthelme 5-m-Anschlussleitung für LED-Streifen, 5-adrig, schwarz	CL-12 49 28	€ 12,95
Schieberegister 74HC595D,112 High Speed CMOS	CL-11 37 21	€ 0,21
Schieberegister High Speed CMOS 74 HC595	CL-00 47 52	€ 0,79
Schieberegister High Speed CMOS 74HC165	CL-02 72 98	€ 0,30
microSD-Kartenadapter MSDA1, Bausatz	CL-13 15 91	€ 7,95