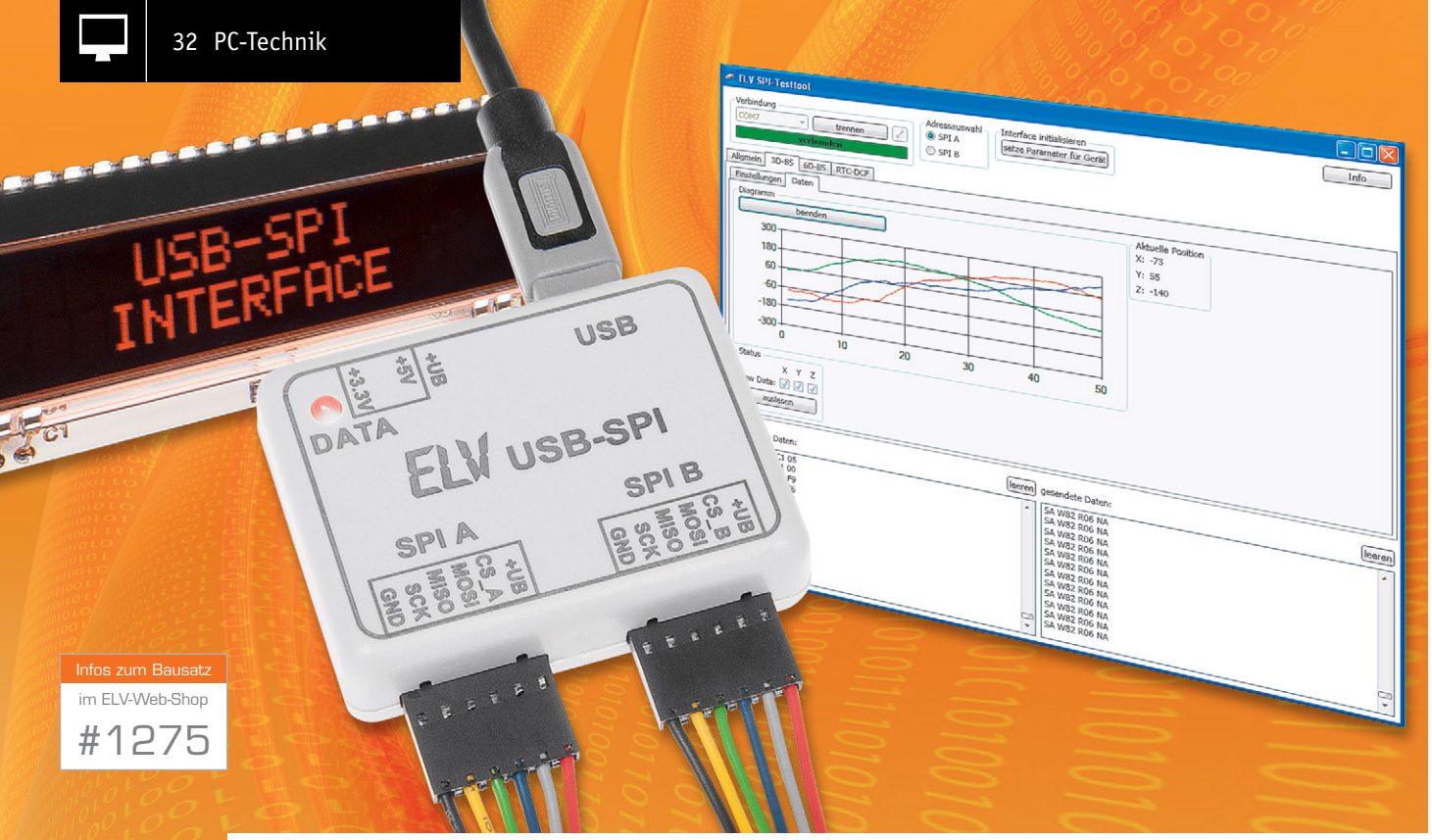


Infos zum Bausatz
im ELV-Web-Shop
#1275

Bequem auf SPI zugreifen –
USB-SPI-Interface

Einfach und schnell messen, steuern, testen und programmieren – mit einem einfachen Bussystem wie dem SPI-Bus ist dies kein Problem, und die große Vielfalt SPI-kompatibler Bausteine, Prozessoren und Geräte macht zahlreiche Lösungen einfach. Mit unserem kleinen Interface kann von einem PC aus via USB mit einfachsten Befehlen direkt auf angeschlossene Geräte oder Bausteine mit SPI-Schnittstelle zugegriffen werden. Durch die Makrofunktion ist das Interface z. B. mit nur einem zusätzlichen IC als Datenlogger für analoge/digitale Signale, Temperaturwerte usw. konfigurierbar.

Angebunden

Wie auch der I²C-Bus ist der SPI-Bus weit verbreitet, er bietet insbesondere den Vorteil der schnelleren Datenübertragung, benötigt aber dafür mehr Leitungen, als es bei I²C der Fall ist.

Technische Daten

Geräte-Kurzbezeichnung:	USB-SPI
Versorgungsspannung:	USB-powered
Stromaufnahme (gesamt):	250 mA max.
Stromaufnahme (eigene):	<50 mA
Ausgangsspannung:	3,3 oder 5 V (wählbar)
Ausgangslast (gesamt):	200 mA max.
Schutzart:	IP20
Umgebungstemperatur:	5 bis 35 °C
Mögliche SPI-Taktfrequenzen:	62,5; 125; 250; 768 kHz; 1; 2; 4 MHz
Leitungslängen:	max. 10 cm
Mögliche Baudraten:	4.800, 9.600, 19.200, **38.400**, 76.800, 250.000, 500.000 Baud
Anzeigeelement:	Duo-LED für Bus-Aktivität und Spannungseinstellung
PC-Anbindung:	USB (Kommunikation über virtuellen COM-Port)
Abmessungen (B x H x T):	39 x 14 x 50 mm
Gewicht:	18 g

Eines der wohl typischsten Beispiele ist die Anbindung von Schieberegistern (74HC595/74HC165), damit lässt sich ein Mikrocontroller mittels weniger Leitungen über den SPI-Bus um eine Vielzahl von I/Os erweitern. Aber auch eine Vielfalt anderer Komponenten lässt sich über SPI ansprechen, seien es EEPROMs oder Flash-Speicher, A/D- bzw. D/A-Wandler, LCD-/LED-Treiber oder analoge Sensoren wie Temperatur-, Luftfeuchtigkeits- oder Bewegungssensoren. Gerade bei den Flash-Speichern ist die höhere Geschwindigkeit des SPI-Busses ein ausschlaggebendes Kriterium.

Natürlich liegt es nahe, solche Bausteine wegen ihrer vielseitigen Einsatzmöglichkeiten auch an einen PC anzubinden, nur dem fehlt dort halt eine dem Anwender zugängliche SPI-Schnittstelle.

Dieses Manko auszugleichen, ist die Aufgabe unseres kleinen Interfaces. Es setzt eine USB-Schnittstelle mittels eines USB-UART-Wandlers und eines kleinen Mikrocontrollers in eine SPI-Schnittstelle um und umgekehrt.

Es sind zwei einzelne Chip-Select-Leitungen vorhanden, um auch zwei Komponenten gleichzeitig am USB-SPI betreiben zu können oder für Geräte, die eine weitere Steuerleitung benötigen. Über einen relativ einfachen Befehlsalgorithmus lassen sich SPI-Komponenten so direkt vom PC aus ansprechen bzw. steuern.

Auf diese Weise lassen sich sehr unkompliziert auch größere Mess-, Regel- und Steueraufgaben mit SPI-kompatiblen Komponenten aufbauen, im Kasten „Elektronikwissen“ werden dazu einige Grundlagen des SPI-Bussystems betrachtet.

Schaltung

Die Schaltung des USB-SPI-Interfaces (Bild 1) ist recht überschaubar, im unteren Teil ist die USB-Schnittstelle mit dem zugehörigen USB-UART-Wandler IC4 zu sehen.

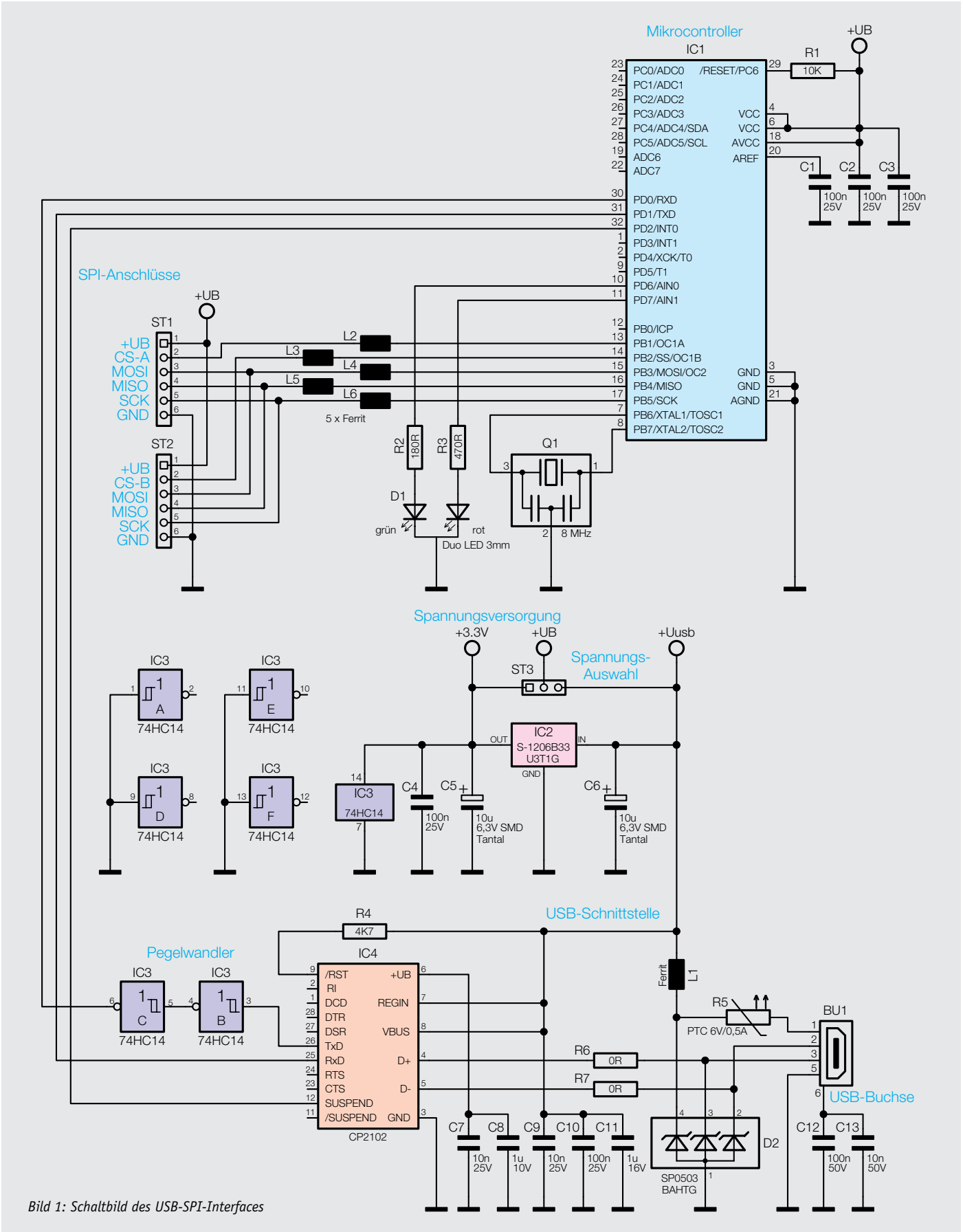


Bild 1: Schaltbild des USB-SPI-Interfaces

Darüber befindet sich mit IC2 ein Linearregler zur Erzeugung einer 3,3-V-Spannung aus der USB-Versorgungsspannung und die Spannungsauswahl für +UB über den Jumper ST3. L1 und die Kondensatoren C5 und C6 dienen zur Filterung und Stabilisierung der Versorgungsspannungen, während R5 den Überstromschutz übernimmt.

Mit ST3 kann zwischen der 3,3- und der 5-V-USB-Spannung gewählt werden, so dass man wahlweise 3,3- oder 5-V-SPI-Slave-Komponenten anschließen kann.

Im oberen Teil des Schaltplans befindet sich der Mikrocontroller IC1, er setzt die Daten vom UART (USB) auf die SPI-Schnittstellen ST1 und ST2 um und umgekehrt, die Ferrite L2 bis L6 dienen dabei als Filter. Q1 stellt einen 8-MHz-Takt für den Mikrocontroller bereit.

Die Duo-LED D1 wird über die Vorwiderstände R2 und R3 direkt vom Mikrocontroller gesteuert, sie dient zum einen zur Anzeige der ausgewählten Spannung (grün für 3,3 V und rot für 5 V), aber auch für die Signalisierung der Datenübertragung.

Der Mikrocontroller misst seine interne Referenzspannung gegenüber der Versorgungsspannung und kann so die ausgewählte Spannung ermitteln und entsprechend die LEDs ansteuern.

IC3 ist zwischen den USB-UART-Wandler IC4 und den Mikrocontroller IC1 geschaltet und sorgt für definierte Pegel je nach Spannungsauswahl. Da der CP2102 z. B. High-Pegel schon ab 3 V definiert, der Mikrocontroller bei 5-V-Spannungsversorgung aber High-Pegel erst ab 3,2 V erkennt, wird der Pegel von IC3 auf 3,3 V angehoben, um auch bei Betrieb mit 5 V die Datenübertragung sicherzustellen.

Nachbau

Der Aufbau des Interfaces ist schnell erledigt, da alle SMD-Bauteile bereits ab Werk bestückt sind.

Es bleibt noch die Bestückung weniger bedrahteter Bauteile. Dabei beginnen wir mit ST1 bis ST3, die wie im Platinenfoto (Bild 2) zu sehen, zu bestücken und auf der Platinen-Lötseite zu verlöten sind. Dabei sind die Lötstellen sorgfältig auszuführen und es ist darauf zu achten, dass die Kunststoffträger der Stiftleisten plan auf der Platine aufliegen.

Abschließend ist die LED D1 zu bestücken, hier ist auf die richtige Ausrichtung anhand des Bestückungsdrucks zu achten. Die LED ist so einzulöten, dass sich eine Einbauhöhe von 11 mm von der Platinenoberfläche bis zur LED-Spitze ergibt. Bild 3 zeigt die so fertig bestückte Platine.

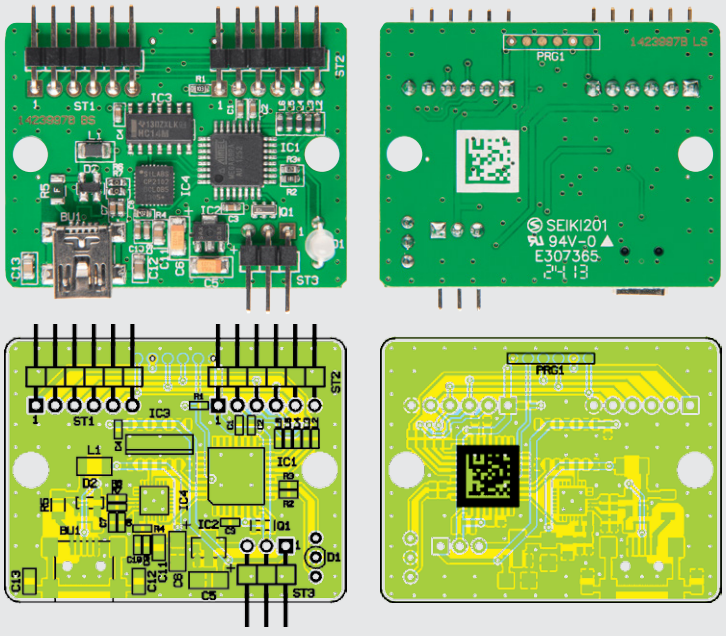


Bild 2: Die komplett bestückte Platine des USB-SPI-Interfaces mit dem zugehörigen Bestückungsplan

Damit ist die Bestückung bereits abgeschlossen und nach der Kontrolle auf Löt- und Bestückungsfehler kann der Einbau ins Gehäuse erfolgen. Dazu wird die Platine in die Gehäuseunterschale eingelegt, so dass die Buchsen exakt in den vorbereiteten Ausschnitten liegen. Nun ist die Gehäuseoberschale aufzulegen und mit den zwei beiliegenden Schrauben zu verschrauben. In Bild 4 ist das fertig montierte Gerät zu sehen. Dieses ist nun bereit zur Inbetriebnahme.

Inbetriebnahme

Dazu wird zunächst der Jumper in die gewünschte Position zur Spannungsauswahl 3,3 oder 5 V auf ST3 gesteckt. Danach kann ein SPI-Gerät angeschlossen werden, gefolgt vom Anschluss an die USB-Buchse.

Das Gerät meldet sich nach der Treiberinstallation (siehe Abschnitt „Installation“) mit USB-SPI an und stellt einen virtuellen COM-Port bereit. Dieser kann

Stückliste

Widerstände:	
0 Ω/SMD/0603	R6, R7
180 Ω/SMD/0603	R2
470 Ω/SMD/0603	R3
4,7 kΩ/SMD/0603	R4
10 kΩ/SMD/0603	R1
Polyswitch/6 V/0,5 A/SMD/1206	R5
Kondensatoren:	
10 nF/SMD/0603	C7, C9
10 nF/SMD/0805	C13
100 nF/SMD/0603	C1–C4, C10
100 nF/SMD/0805	C12
1 µF/SMD/0603	C8
1 µF/SMD/0805	C11
10 µF/6,3 V/Tantal/SMD	C5, C6
Halbleiter:	
ELV131189/SMD	IC1
S-1206B33-U3T1G/SMD	IC2
74HC14/SMD	IC3
ELV131305/SMD/USB-Controller	IC4
SP0503BAHTG	D2
Duo-LED/rot/grün/3 mm	D1
Sonstiges:	
Keramikschwinger, 8 MHz, SMD	Q1
Chip-Ferrit, 1206, 80 Ω bei 100 MHz	L1
Chip-Ferrit, 0603, 60 Ω bei 100 MHz	L2–L6
USB-B-Buchse, mini, 5-polig, winkelprint, liegend, SMD	BU1
Stiftleiste, 1x 6-polig, winkelprint	ST1, ST2
Stiftleiste, 1x 3-polig, winkelprint	ST3
2 flexible Leitungen mit 1 Crimp-Buchse, 6-polig, 20 cm	
1 Jumper ohne Grifflasche, geschlossene Ausführung	
1 Kunststoffgehäuse, komplett, bearbeitet und bedruckt	
1 USB-Kabel (Typ A auf Typ B mini), 2 m, schwarz	
1 Mini-CD	

nun mit einem beliebigen Terminal-Programm, z. B. HTerm, gesteuert werden.

Für den Anschluss eigener SPI-Slave-Komponenten an das Interface liegen dem Bausatz zwei Buchsen mit fertig konfektionierten Kabeln bei. Diese haben auf der Oberseite einen Verpolungsschutz, der genau in die Gehäuseöffnungen passt. Die Belegung der Adern ist direkt auf das Gehäuse aufgedruckt (rotes Kabel = +UB [3,3 V/5 V], schwarzes Kabel = GND).

Hierüber kann ein einfacher Anschluss an das Interface erfolgen und der Kommunikation zwischen PC und SPI-Geräten steht nichts mehr im Wege.

Des Weiteren befindet sich auf der beiliegenden CD eine Demo-Software mit Quellcode für das USB-SPI-Interface, welche z. B. für den 3-Achsen-Bewegungssensor fertige Funktionen bietet und so den Einstieg etwas erleichtert.

Installation und Bedienung

Vor dem Anschluss des USB-SPI-Interfaces ist der Treiber wie im Folgenden beschrieben zu installieren. Dabei sollte man einmal kurz überprüfen, ob unter [1] vielleicht eine neuere Version zum Download bereitsteht, welche man dann der CD-Version vorziehen sollte.

1. Silabs-VCP-Treiber (Virtual-COM-Port) installieren
2. USB-SPI-Interface über das beiliegende USB-Kabel an den PC anschließen (vorerst ohne angeschlossene SPI-Hardware)
3. Das Interface wird vom Betriebssystem als neues Gerät erkannt, es öffnet sich der Installationsassistent, dessen Anweisungen zu befolgen sind.
4. Nun ist im Windows-Gerätemanager zu prüfen, welcher COM-Port dem Gerät zugewiesen wurde. Dieser lässt sich im Gerätemanager über: „Eigenschaften"-> „Erweitert..." ändern, siehe dazu Bild 5.
5. Schließlich ist ein beliebiges Terminalprogramm (z. B. HTerm [2]) zu starten, der zugewiesene COM-Port auszuwählen und mit folgenden Einstellungen zu öffnen (Bild 6): 38.400 Bit/s, 8 Datenbits, 1 Stoppbit, keine Parität, keine Flusssteuerung (Handshake).
6. Nun kann man eigene SPI-Slave-Geräte anschließen und die Kommunikation starten.

Die Kommunikation mit den SPI-Geräten

Wie in „Elektronikwissen" beschrieben, wird in der Regel bei der Kommunikation mit SPI-Komponenten ein Chip-Select-Signal benötigt. Für die beiden SPI-Anschlüsse steht deshalb jeweils eine Chip-Select-Leitung A, B zur Verfügung.

Das Setzen des Chip-Selects erfolgt mit dem ASCII-Zeichen „S", wie der Tabelle 1 zu entnehmen ist, danach folgt die Auswahl des Kanals A oder B oder beider mit AB. Die Abwahl des Chip-Selects erfolgt mit dem Zeichen „N", gefolgt von dem gewünschten Kanal.

Anhand des Beschleunigungssensors 3D-BS soll der Ablauf der Kommunikation näher beschrieben werden.

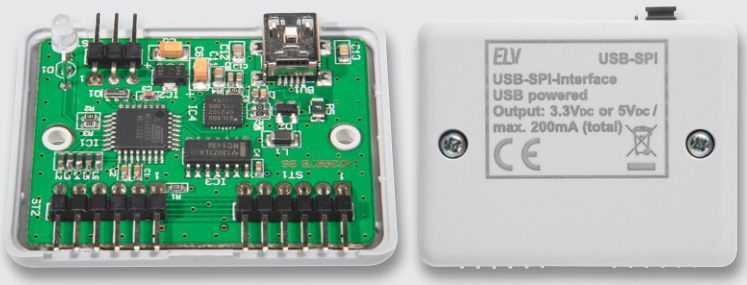


Bild 3: Die mit Stiftleisten und der LED bestückte Platine wird in die Gehäuseunterschale eingelegt und die beiden Gehäuseschalen werden miteinander verschraubt.

Bild 4: Das fertig montierte Gerät. Rechts sind deutlich die Aussparungen für die vorkonfektionierten Stecker zu sehen, die ein polrichtiges Stecken erzwingen.

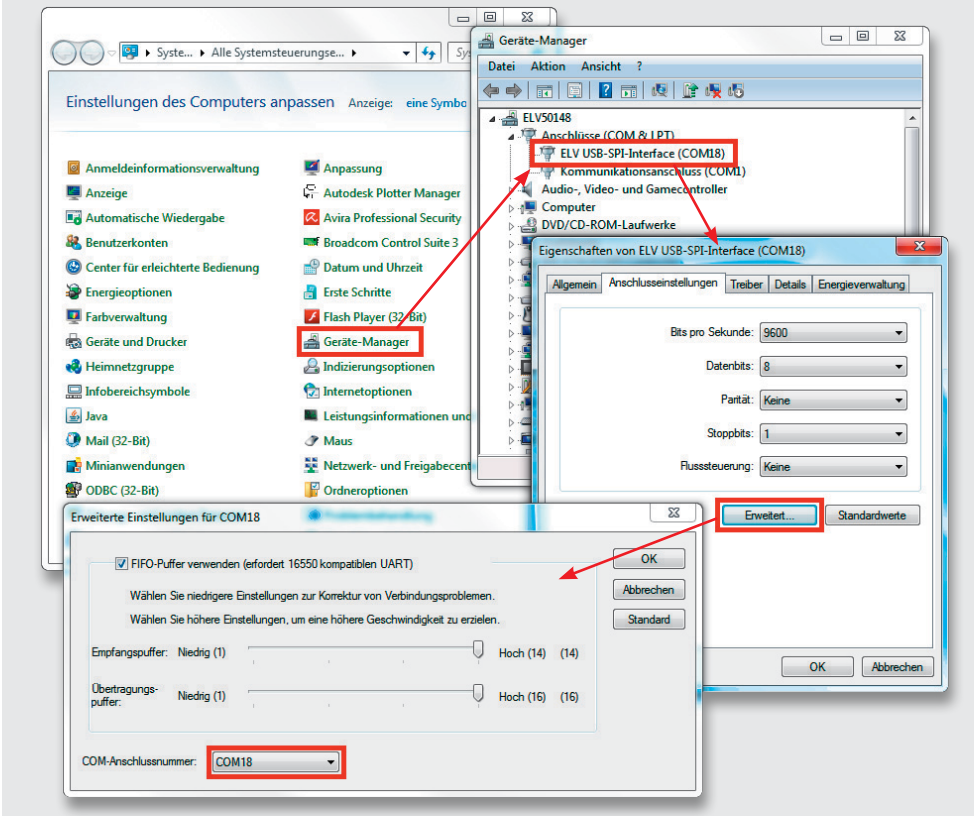


Bild 5: So findet man den virtuell gebildeten COM-Port des Interfaces und die zugehörigen Kommunikationseinstellungen.

Wie in Bild 7 zu sehen, wird bei diesem Beispiel der SPI-Kanal A verwendet.

Zu Beginn der Kommunikation mit einer SPI-Komponente sind die SPI-Einstellungen zu beachten, auch hier sei auf „Elektronikwissen“ verwiesen.

Wichtiger Hinweis:

Die Leitungen, die an den mit „SPI“ beschrifteten Schnittstellen angeschlossen werden, dürfen eine Länge von 10 cm nicht überschreiten. Gleichzeitig gilt: Je höher die SPI-Taktrate eingestellt wird, desto kürzer sollten die Leitungen sein, um Datenfehler zu vermeiden.

Beim 3D-BS sind CPOL und CPHA jeweils auf 1 zu setzen, dies geschieht mit den Zeichenfolgen „Y11“ und „Y21“, die Bitreihenfolge DORD ist mit der Zeichenfolge „Y30“ auf 0 zu setzen.

Da der 3D-BS nur einen Takt von maximal 100 kHz unterstützt, wird mittels „T00625“ die Busgeschwindigkeit auf 62,5 kHz eingestellt.

Für den ersten Test wird die Chip-ID des 3D-BS ausgelesen, dazu ist das Register 0x00 zu lesen. Dabei muss bei einem Lesezugriff in der Registeradresse das oberste Bit gesetzt werden, so dass sich die Adresse 0x80 ergibt.

Ein Lesezugriff besteht zum einen aus einem Schreibbefehl, bei dem die zu lesende Adresse der Komponente mitgeteilt wird, zum anderen dem eigentlichen Lesebefehl:

SA	W80	R01	NA
Chip auswählen	Adresse schreiben	1 Byte lesen	Chip abwählen

Wir erhalten auf diesen Befehl ein Byte (2 ASCII-Zeichen) als Antwort.

Elektronikwissen

SPI und Übertragungseinstellungen

Das Serial Peripheral Interface (SPI) ist ein synchroner serieller Datenbus, entwickelt von Motorola, wobei jedoch lediglich die Hardware-Funktionsweise beschrieben wurde. Ein Software-Protokoll oder gar ein Standard existieren nicht, ebensowenig wie Patente oder Lizenzen, was den Bus für jeden frei nutzbar macht.

SPI ist als Master-Slave-Bus ausgelegt, d. h. ein Master übernimmt bei der Datenübertragung die Steuerung, wählt einen Slave an und treibt den Takt.

Die Datenübertragung kann gleichzeitig in beide Richtungen erfolgen, also Voll-Duplex.

Der Bus besteht zunächst aus 3 Datenleitungen:

- SDO (Serial Data Out) bzw. MISO
- SDI (Serial Data In) bzw. MOSI
- Taktleitung SCK (Serial Clock)

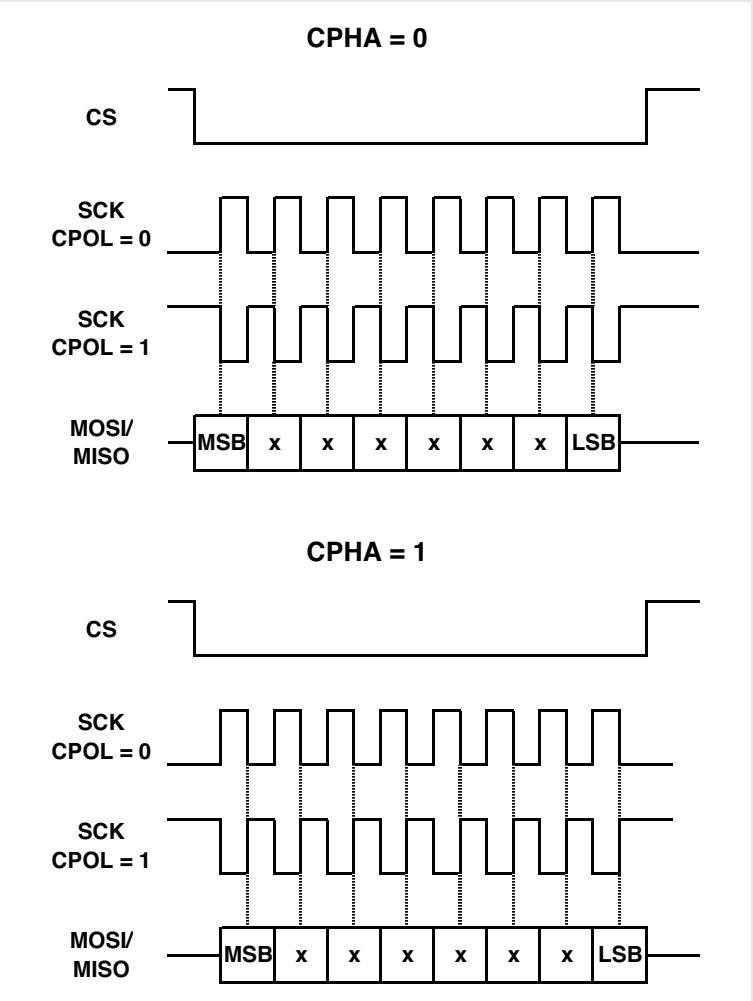
Zusätzlich zu den Datenleitungen wird für jeden Slave eine Select-Leitung genutzt, um einen Slave auszuwählen, diese wird häufig als Slave-Select oder Chip-Select bezeichnet.

Sind an den SPI mehrere Slaves angeschlossen, erfolgt über die Select-Leitungen die Auswahl des anzusprechenden Slaves. Dabei wird üblicherweise mit einem Low-Pegel die Auswahl aktiviert und mit einem High-Pegel deaktiviert.

Bei der Datenübertragung können die Daten zu unterschiedlichen Zeitpunkten vom Bus übernommen werden, dies wird mittels der folgend aufgeführten Einstellungen festgelegt. In der Praxis haben sich dabei folgende Modi durchgesetzt:

- Mittels CPOL wird die Polarität des Taktes bestimmt, dabei ist bei CPOL = 0 der Takt im Idle Low und bei CPOL = 1 im Idle High.
- Über die Einstellung CPHA wird bestimmt, zu welcher Flanke die Daten vom Bus übernommen werden. Bei CPHA = 0 bei der ersten Flanke des Taktsignals (je nach Polarität fallend/steigend) oder bei CPHA = 1 bei der zweiten Flanke des Taktsignals (je nach Polarität fallend/steigend) nach dem Chip-Select.

Bei der jeweiligen anderen Flanke werden die Daten an den Bus angelegt, so dass bis zum Einlesen genügend Zeit zur Stabilisierung der Signale bleibt (siehe Grafik).



In dem hier dargestellten Beispiel wird zuerst das MSB übertragen und das LSB zuletzt, diese Reihenfolge lässt sich aber auch vertauschen, so dass zuerst das LSB und zuletzt das MSB übertragen wird. Dazu ist die Einstellung DORD auf 1 zu setzen.

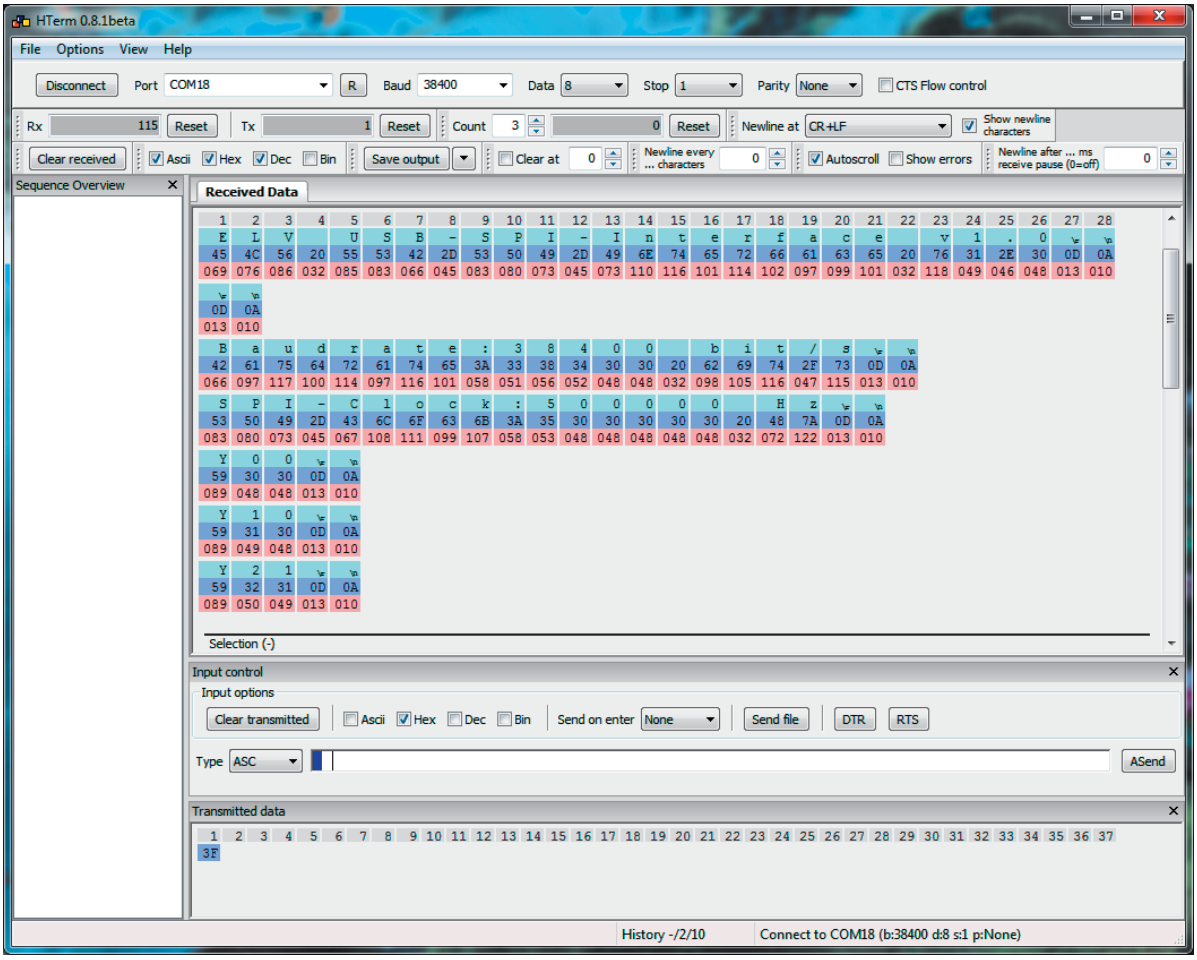


Bild 6: Die Kommunikationseinstellungen für das Interface in HTerm

Tabelle 1

Befehle für die SPI-Kommunikation

ASCII-Zeichen	Folgebyte(s)*/Zeichen	Funktions-Beschreibung
S	‚A' und/oder ‚B'	Chip-Select A und/oder B setzen (aktiv-low)
N	‚A' und/oder ‚B'	Chip-Select A und/oder B zurücksetzen (high)
W	Byte1 Byte2 Byte3...	schreibt „Byte1, Byte2, Byte3..." auf dem SPI-Bus
R	Byteanzahl	liest x Datenbyte (1...255) vom SPI-Bus
:		wartet mit der Ausführung der nachfolgenden Befehle bis zum nächsten Zeilenumbruch (0x0D oder 0x0A); ist sinnvoll, wenn das Terminal-Programm jedes Zeichen sofort nach der Eingabe überträgt; Ausführung erst nach Abschluss mit Eingabetaste
L	Byte1 Byte2	fügt eine Wartepause von 1 bis 65.535 ms (0001...FFFF) in Hex-Schreibweise (16 Bit) in die Befehlsausführung ein (z. B. innerhalb von Makros)

** Jedes Byte (Hexadezimal) wird mit 2 ASCII-Zeichen geschrieben, z. B.: 0x1F = 1F.*

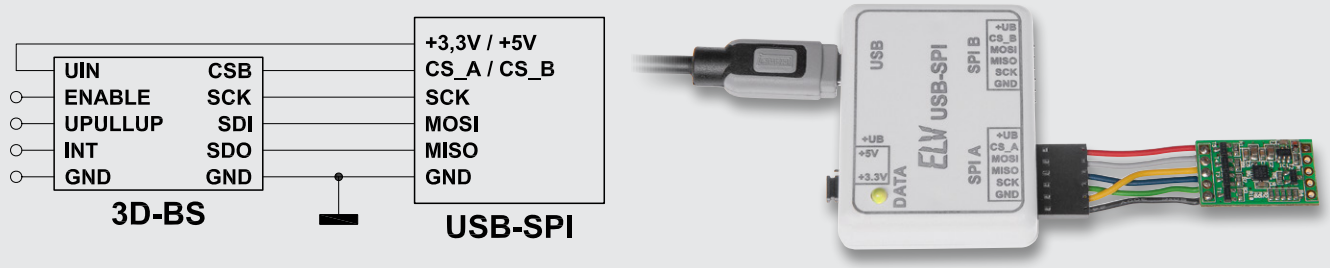


Bild 7: Anwendungsbeispiel zum Betrieb des 3-Achsen-Beschleunigungssensors 3D-BS am USB-SPI-Interface

Die Messwerte des 3D-BS sind ab Register 0x02 enthalten und umfassen 6 Byte.

Daraus ergibt sich zum Auslesen der Messwerte folgende Zeichenfolge:

SA	W82	R06	NA
Chip auswählen	Adresse schreiben	6 Byte lesen	Chip abwählen

Die Antwort besteht z. B. aus den folgenden ASCII-Zeichen:

A2 EA F5 FF 02 46

Mittels der in Tabelle 2 aufgeführten Kommentarfunktionen lässt sich die Rückgabe deutlich übersichtlicher gestalten:

SA W82 [Wert-X:] R02 . [Wert-Y:] R02 . [Wert-Z:] R02 . NA

Die besser verständliche Rückgabe sieht in diesen Fall folgendermaßen aus:

Wert-X: A2EA
Wert-Y: F5FF
Wert-Z: 0246

Mittels des Makrospeichers des USB-SPI können diese Daten auch zyklisch abgefragt werden, so dass ein Datenlogger entsteht. Die Daten werden dann automatisch vom 3D-BS erfasst, vom USB-SPI abgefragt und zum PC übertragen.

Wird das Gerät vom USB-Port getrennt und später neu verbunden, so nimmt der Datenlogger selbstständig erneut die Arbeit auf. Für die Datenaufnahme benötigt man lediglich ein einfaches Terminal-Programm wie z. B. HTerm, das die Daten entgegennimmt und abspeichert. Anschließend können die Daten mit MS Excel oder ähnlichen Programmen ausgewertet und visualisiert werden.

Ein einfaches und sehr kurzes Makro für solch eine Datenloggerfunktion lautet z. B. folgendermaßen:

SA	W82	R06	NA	L0200	>00
Chip auswählen	Adresse schreiben	6 Byte lesen	Chip abwählen	Warte 512 ms (0x200=512)	Starte die Ausführung des Makros ab Adresse 00

Jetzt muss dieses Makro noch in den Makrospeicher ab Adresse 00 geschrieben werden, was mit dem aus Tabelle 3 entnommenen Befehl V00{...} erfolgt:

V00{ SA W82 R06 NA L0200 >00 }

Sollen die Messergebnisse als semikolongetrennte Werte direkt in Excel eingelesen werden, so ist zuerst mit dem Konfigurationsbefehl Y01 aus Tabelle 3 der automatische Zeilenumbruch nach jedem Datenbyte abzuschalten. Anschließend müssen Semikolons zwischen die Datenbytes eingefügt und jede Messperiode mit einem manuell eingefügten Zeilenumbruch (Punkt) abgeschlossen werden. Das Ergebnis ist:

Y01
V00{ SA W82 R02;R02;R02. L0200 >00 }

Die Textausgabe sieht dann (mit konstanten Messergebnissen) folgendermaßen aus:

A2EA; F5FF; 0246
A2EA; F5FF; 0246
A2EA; F5FF; 0246
A2EA; F5FF; 0246

Weitere Beispiele

Die hier in der Folge kurz aufgeführten Beispiele zeigen einige weitere (Standard-)Anwendungen für das Interface. Genaue Ausführungen zur Konfiguration und Behandlung der jeweiligen Baugruppen sind deren Anleitungen zu entnehmen.

Beispiel 6-Achsen-Beschleunigungssensor 6D-BS

Der Anschluss des Sensors ist in Bild 8 zu sehen.
Die zugehörigen SPI-Einstellungen lauten hier:

Y11
Y21
Y30
T01250

Beim 6D-BS müssen, damit der Chip aktiv wird, vorher einige Einstellungen vorgenommen werden, diese sind dem Datenblatt des 6D-BS zu entnehmen, hier folgt nur ein Beispiel:

Konfiguration: SA W60 97 00 00 08 00 00 NA SB W60 F7 00 00 30 00 NB

Chip-ID auslesen: SB W8F R01 NB
Werte des Gyroskops auslesen: SB WE8 R06 NB
Werte einzeln auslesen: SB WA8 R01 NB SB WA9 R01 NB SB WAA R01 NB ...

Hinweise zu Registeradressen:
7. Bit (MSB): Write = 0; Read = 1
6. Bit: gleiche Registeradr. = 0; autoincrement Adr. = 1
5. bis 0. Bit: Registeradr.

Beispiel RTC-DCF

Der Anschluss des Echtzeituhr-DCF-Bausteins ist in Bild 9 zu sehen.

Hier ist unbedingt zu beachten, dass dieser Baustein nur mit 3,3 V zu betreiben ist!

Tabelle 2

Kommentarbefehle für die Befehls- und Rückgabewerte

ASCII-Zeichen	Funktions-Beschreibung
.	ein Punkt bewirkt einen Zeilenumbruch (0x0D 0x0A) in der Rückgabe
,	fügt ein Komma in die Rückgabe ein (für kommagetrennte Daten für Excel o. Ä.)
;	fügt ein Semikolon in die Rückgabe ein (für semikolongetrennte Daten für Excel)
[...]	ASCII-Zeichen zwischen eckigen Klammern werden zum PC zurückgegeben –> zum Kommentieren von Rückgabewerten, mögliche Steuerzeichen: \r (Carriage Return), \n (Line Feed), \t (Horizontal Tab)
(...)	ASCII-Zeichen zwischen runden Klammern werden ignoriert –> zum Kommentieren von Befehlsanweisungen
Leerstelle	Leerstellen in den Anweisungen werden ignoriert (ausgenommen innerhalb von eckigen Klammern)

Die zugehörigen SPI-Einstellungen lauten hier:
Y10
Y21
T50000

Lesen aller Register:
SA W40 R0F NA
Schreiben von Uhrzeit und Datum:
SA W00 43(43 Sekunden) 15(15 Minuten) 08(08 Stunden) 01(Montag) 25(Tag 25.) 04(Monat 04.) 14(Jahr 14) NA

Bild 8: Anwendungsbeispiel zum Betrieb des 6-Achsen-Beschleunigungssensors 6D-BS am USB-SPI-Interface

Tabelle 3

Befehlsübersicht zur Konfiguration des USB-SPI-Interfaces

ASCII-Zeichen	Folgebyte(s)*/Zeichen		Funktions-Beschreibung
>	Makroadresse (1 Byte)		startet Makro-Ausführung an der Makro-Speicheradresse
<			beendet Makro-Ausführung und wartet auf neue Anweisungen vom PC
V	Makroadresse { Zeichen1 Zeichen2... }		schreibt die ASCII-Zeichen (Befehle und Daten) zwischen den geschweiften Klammern in den Makrospeicher ab der übergebenen Makroadresse; Makrospeicher löschen mit: V00{} (Speicher wird mit Leerzeichen [Hex:0x20] überschrieben)
U			Inhalt des Makrospeichers (256 ASCII-Zeichen) wird vollständig ausgegeben (zum PC)
?			Systemstatus und Einstellungen werden ausgegeben (zum PC)
T	SPI-Taktrate (5 Zeichen)		SPI-Bustakt einstellen: 00625 = 62,5 kHz, 01250 = 125 kHz, 02500 = 250 kHz, 05000 = 500 kHz, 10000 = 1 MHz, 20000 = 2 MHz, 40000 = 4 MHz
X	Baudrate (4 Zeichen)		COM-Port Baudrate: 0048 = 4800 bit/s, 0096 = 9600 bit/s, 0192 = 19.200 bit/s, 0384 = 38.400 bit/s #, 0768 = 76.800 bit/s, 2500 = 250.000 bit/s, 5000 = 500.000 bit/s (diese Einstellung wird erst nach einem Neustart wirksam)
Y	0	0#	dem letzten Datenbyte, das der Master aus dem Slave liest, folgt ein Zeilenumbruch (0x0D 0x0A) in der Rückgabe zum PC
		1	Daten, die der Master aus dem Slave ausliest, folgt kein Zeilenumbruch in der Rückgabe zum PC
	1	0#	CPOL=0: Polarität des Taktes (in Idle Low-Pegel)
		1	CPOL=1: Polarität des Taktes (in Idle High-Pegel)
	2	0#	CPHA=0: Zeitpunkt der Datenübernahme vom Bus (1. Flanke)
		1	CPHA=1: Zeitpunkt der Datenübernahme vom Bus (2. Flanke)
	3	0#	DORD=0: MSB wird zuerst übertragen, LSB zuletzt
		1	DORD=1: LSB wird zuerst übertragen, MSB zuletzt
	4	0#	jeder Daten-Rückgabe (2 Zeichen) folgt ein Leerzeichen (0x20)
		1	einer Daten-Rückgabe zum PC folgt kein Leerzeichen
	5	0#	nach einem Reset das Makro ausführen, wenn eines im Makrospeicher steht
		1	nach einem Reset kein Makro ausführen
	6	0#	die Makroadresse (nach > und V) wird mit 2 und die Wartepause (nach L) wird mit 4 ASCII-Zeichen in Hexadezimalschreibweise angegeben
		1	die Makroadresse (nach > und V) wird mit 3 und die Wartepause (nach L) wird mit 5 ASCII-Zeichen in Dezimalschreibweise angegeben
	7	0#	gelesene Daten werden als ASCII-Zeichen im Hexadezimalformat zum PC gesendet
		1	gelesene Daten werden als ASCII-Zeichen im Dezimalformat (ohne führende 0) zum PC gesendet
Z	4B		startet das USB-I²C-Interface neu (Reset); der Inhalt des Makrospeichers bleibt erhalten
	AA		Konfiguration, Baudrate und Bustakt auf Auslieferungszustand zurücksetzen, Makrospeicher löschen und USB-I²C-Interface neu starten

** Jedes Byte (Hexadezimal) wird mit 2 ASCII-Zeichen geschrieben, z. B.: 0x1F = 1F.*
 # Standardwert im Auslieferungszustand

Die Zahlen für Datum und Zeit werden in Zehner- und Einerstellen aufgetrennt und im „high“ (Zehner)/ „low“(Einer) Nibble eines Byte übertragen. In Tabelle 4 sind die Register des RTC zusammengefasst.

Beispiel Display mit DOG-M

Für den Anschluss eines DOG-M-Displays (Bild 10) wird zusätzlich eine zweite Steuerleitung neben dem Chip-Select benötigt, um zwischen Daten und Befehlen unterscheiden zu können, dafür wird hier die zweite Chip-Select-Leitung verwendet.

Die zugehörigen SPI-Einstellungen lauten hier:

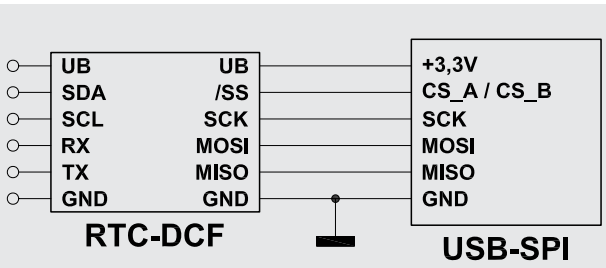
Y11
Y21
Y30

Schreiben auf dem Display:

SAB (init) W39 1C 52 69 74 0c 06 01 L0002 NB (DOG-M) W44 4f 47 2d 4d (@) W40 (USB-SPI) W55 53 42 2d 53 50 49 NA

Erläuterungen:

SAB: DOG-M auswählen und Befehlsmodus aktivieren
(init) W39 1C 52 69 74 0c 06 01 L0002:
Initialisierung des DOG-M und Setzen der Einstellungen
(Näheres bitte dem Datenblatt zum DOG-M entnehmen)
NB: Befehlsmodus deaktivieren, nun folgen Daten/Text.
(DOG-M) W44 4f 47 2d 4d (@) W40 (USB-SPI) W55 53 42 2d 53 50 49:
Daten/Text zum Display senden.
NA: DOG-M abwählen.



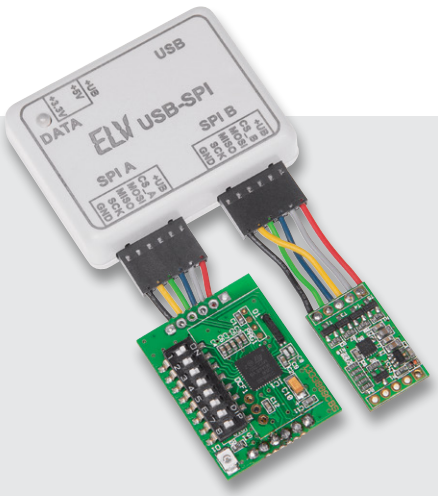
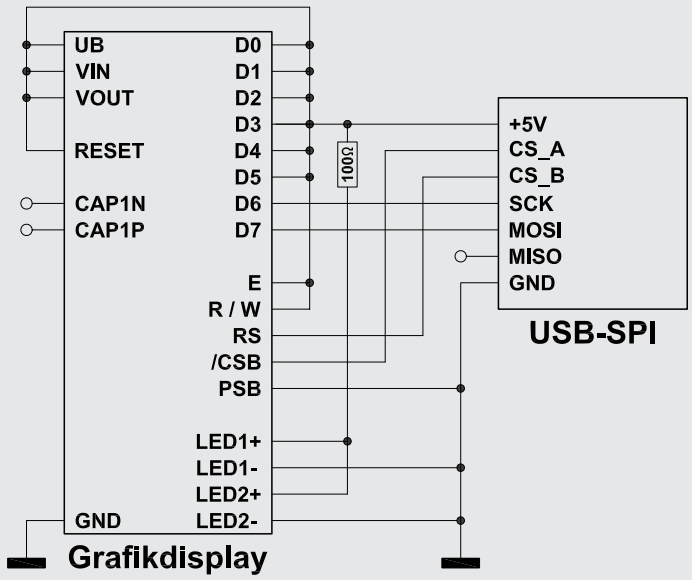


Bild 9: Anwendungsbeispiel zum Betrieb des Echtzeituhr-DCF-Moduls RTC-DCF am USB-SPI-Interface. Im Foto ist an Anschluss B zusätzlich ein 3D-BS angeschlossen.



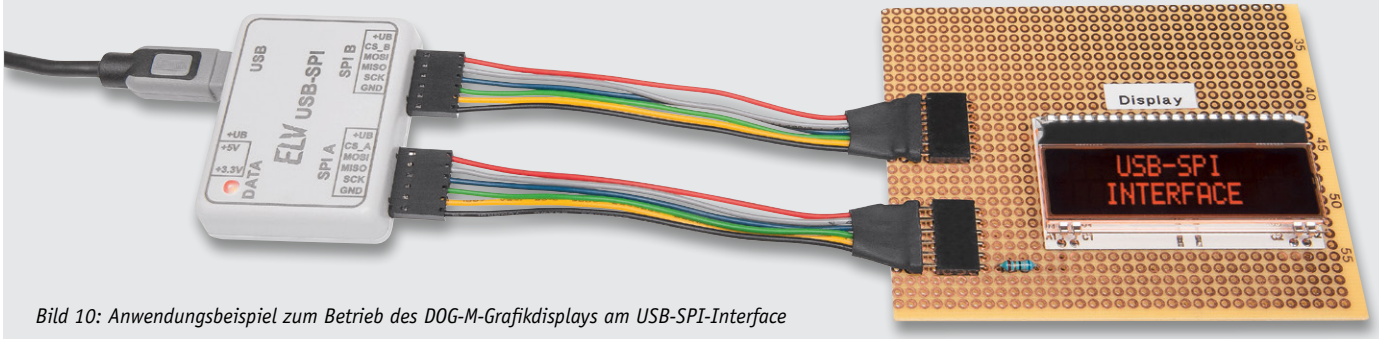


Bild 10: Anwendungsbeispiel zum Betrieb des DOG-M-Grafikdisplays am USB-SPI-Interface

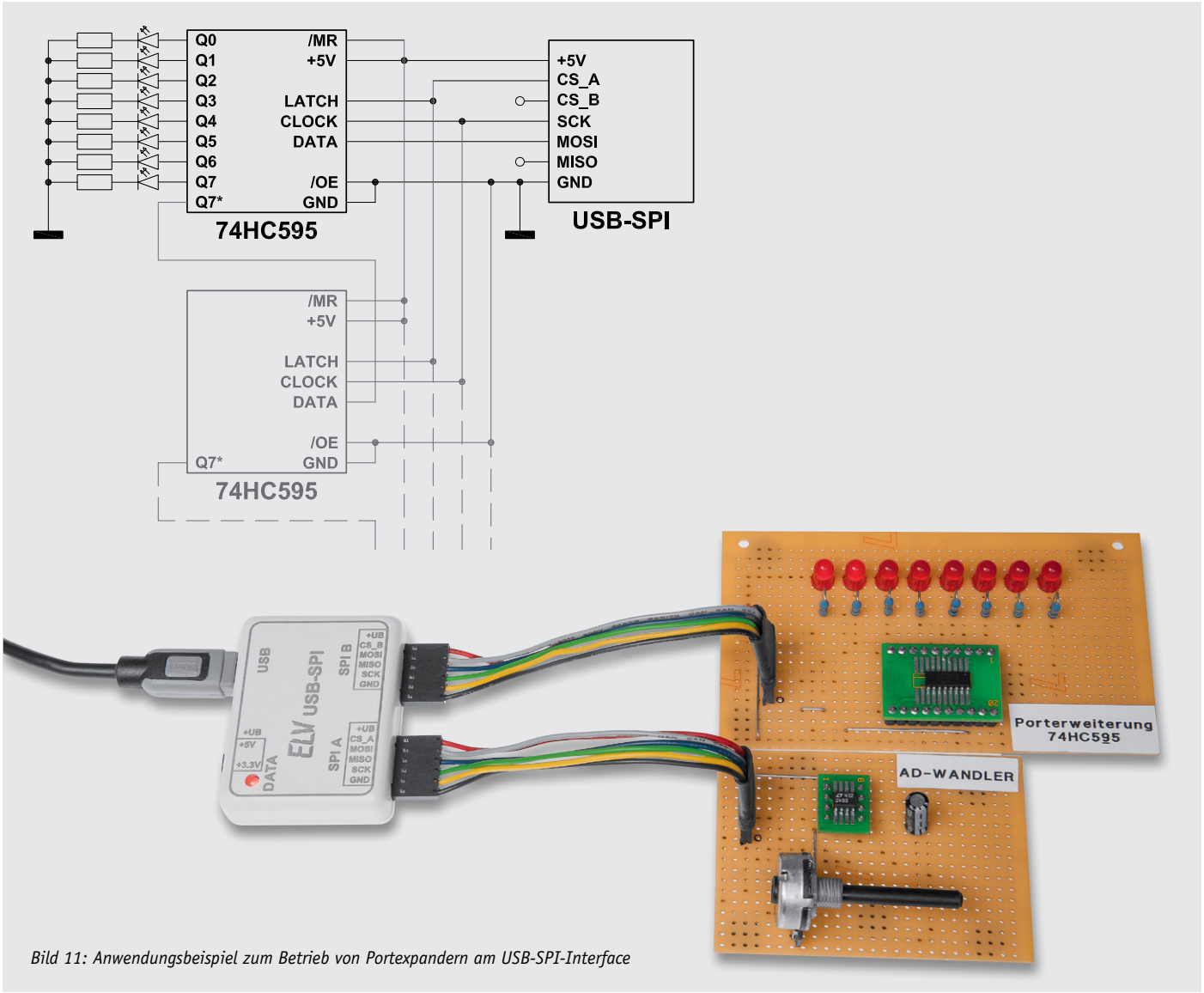


Bild 11: Anwendungsbeispiel zum Betrieb von Portexpandern am USB-SPI-Interface

Beispiel Port-Erweiterung mit 74HC595

Mittels eines 74HC595-Schieberegisters lassen sich über SPI weitere Ausgänge erschaffen, hier als Beispiel für ein Lauflicht (Bild 11).

Da es sich beim 74HC595 um ein Schieberegister handelt, lassen sich beliebig viele dieser Bausteine hintereinanderschalten, um so über nur wenige Leitungen viele Ausgänge anzusteuern (in Bild 11 grau dargestellt).

Bei dem Schieberegister werden die Daten zuerst komplett geschrieben und dann mittels eines Impulses auf dem LATCH-Eingang (Chip-Select) übernommen. Hier dient der Chip-Select nicht, wie bei den anderen Beispielen, zur Anwahl des Schieberegisters, sondern übernimmt die Funktion zum Triggern, wann die Daten aus dem Schieberegister zu den Ausgängen übernommen werden sollen.

Tabelle 4

Registertabelle des RTC-DCF

Adresse	Name	Bits							
		D7	D6	D5	D4	D3	D2	D1	D0
0h	Sekunde	–	ST[2:0]			SU[3:0]			
1h	Minute	–	MNT[2:0]			MNU[3:0]			
2h	Stunde	–	–	HT[1:0]		HU[3:0]			
3h	Wochentag	–	–	–	–	–	WU[2:0]		
4h	Tag	–	–	DT[1:0]		DU[3:0]			
5h	Monat	–	–	–	MT	MU[3:0]			
6h	Jahr	YT[3:0]				YU[3:0]			
7h	Alarm Minute	–	AMNT[2:0]			AMNU[3:0]			
8h	Alarm Stunde	–	–	AHT[1:0]		AHU[3:0]			
9h	Alarm Wochentag	–	AWSU	AWSA	AWFR	AWTH	AWWE	AWTU	AWMO
Ah	Periodischer Interrupt	–	–	–	–	–	PIM[2:0]		
Bh	Alarm-Config-Register	–	–	–	–	–	AILED	AIE	–
Ch	Per.-Int.-Config-Register	–	–	–	–	–	PILED	AIE	–
Dh	DCF77-Config-Register	–	–	–	–	–	DCFLED	DCFIE	DCFE
Eh	Status-Register	–	–	–	–	–	AIF	PIF	DCFIF
Fh	–	–	–	–	–	–	–	–	–

Die zugehörigen SPI-Einstellungen lauten hier:
Y11
Y21
Y30

Ersten Ausgang einschalten:
W01 SA NA

Bei mehreren Schieberegistern hintereinander:
0x01 erster Ausgang vom ersten Schieberegister und 0x02 zweiter Ausgang vom zweiten Schieberegister setzen:
W01 02 ... SA NA
(...: für weitere Schieberegister, beliebig erweiterbar)

PC-Software

Nachdem nun einige Beispiele zur Ansteuerung verschiedener Komponenten mit einem Terminalprogramm vorgestellt wurden, kommen wir zu der Demoanwendung, welche den ersten Einstieg noch erleichtert.

Nach dem Starten des Programms erhalten wir die in Bild 12 abgebildete Ansicht.

Im oberen linken Bereich sind die Verbindungseinstellungen zu finden. Hier sind der COM-Port, die COM-Port-Einstellungen wie Baudrate auszuwählen und die Verbindung herzustellen bzw. zu trennen.

Im Reiter „Allgemein" sind nun Funktionen, die allein das USB-SPI-Interface betreffen, aufrufbar.

Ganz oben kann der Status abgefragt oder eigene Befehle an das USB-SPI gesendet werden.

Darunter kann man die Baudrate und die SPI-Taktgeschwindigkeit einstellen. Bei Änderungen der Baudrate ist ein Reset am Gerät auszuführen, bevor die Einstellungen wirksam werden. Änderungen am SPI-Takt werden sofort übernommen.

Im Bereich „Makrospeicher" kann der Inhalt des Makrospeichers ausgegeben oder gelöscht werden. Zudem können hier Makros gestartet und gestoppt werden.

Im Bereich „Reset" kann das Gerät neu gestartet, ein kompletter Werksreset ausgeführt oder nur die Y-Parameter zurückgesetzt werden.

Unten findet man schließlich zwei Bereiche zur Anzeige der gesendeten und der empfangenen Daten.

In den weiteren Reitern sind die Funktionen für die verschiedenen Beispielgeräte verfügbar.

Bei der Adressauswahl müssen die SPI-Kanäle A oder B ausgewählt werden, und über den Button „Setze Parameter für Gerät" können die SPI-spezifischen Einstellungen (CPOL, CPHA) gesetzt werden, diese sind für die Geräte schon fest in der PC-Software hinterlegt.

Die weiteren Funktionen sind geräteabhängig, beim 3D-BS und 6D-BS sollte die Beschreibung aus den Bedienungsanleitungen der Geräte entnommen werden. Dort werden die Funktionen zwar für das USB-I²C-Interface erläutert, sind hier jedoch auf dieselbe Weise anzuwenden. Bild 13 zeigt die Programmoberfläche für das 3D-BS.

Für das RTC-DCF-Modul (Bild 14) war bislang noch keine PC-Software zur Konfiguration verfügbar, deshalb sollen die Funktionen hier kurz erläutert werden. Die genaue Beschreibung der Funktionen ist der Anleitung des RTC-DCF zu entnehmen.

Im Bereich „Uhrzeit" kann entweder die Uhrzeit – durch Klicken in das Zeitfeld – neu gestellt und mittels „setze Zeit" übertragen oder die Uhrzeit aus dem RTC-Baustein mittels „lese Zeit" ausgelesen werden.

Bei dem RTC-DCF lassen sich verschiedene Interrupts auslösen, diese können einen Ausgang schalten und optional zusätzlich die LED auf dem Modul leuchten lassen.

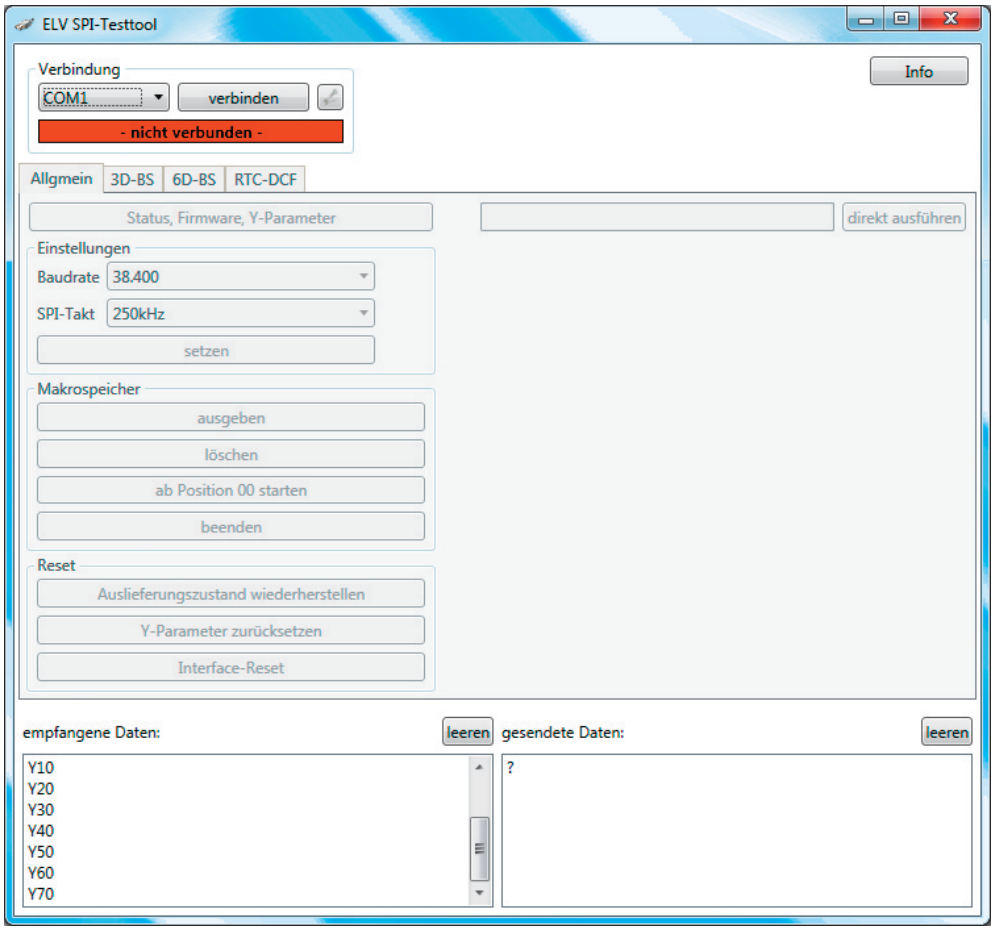


Bild 12: Die Programmoberfläche des zum USB-SPI-Interface gehörenden Konfigurations- und Kommunikationsprogramms

Die aktiven Interrupts lassen sich im Bereich Interrupt-Flags auslesen, während die Konfiguration in den einzelnen Bereichen erfolgt.

DCF:
Beim RTC-DCF besteht die Möglichkeit, ein DCF-Modul anzuschließen, dieses muss aktiviert werden und kann dann bei DCF-Empfang einen Interrupt auslösen.

Alarm:
Mit der Alarmfunktion lässt sich zu einer Uhrzeit an bestimmten Wochentagen ein Interrupt auslösen (z. B. für einen Wecker).

Periodischer Interrupt:
Damit lässt sich von RTC-DCF ein periodisch wiederkehrender Interrupt erstellen, wobei die Frequenz wählbar ist.

Die jeweiligen Einstellungen dazu lassen sich sowohl auslesen als auch setzen. Zusätzlich können die Uhrzeit und die Interrupt-Flags zyklisch von der PC-Software ausgelesen werden, das Intervall ist dabei einstellbar.

Mit dem PC-Programm steht somit ein leistungsfähiges Konfigurations- und Kommunikationswerkzeug für die Arbeit mit dem USB-SPI-Interface zur Verfügung, das den Umgang mit dem vielseitig einsetzbaren Interface sehr einfach macht.

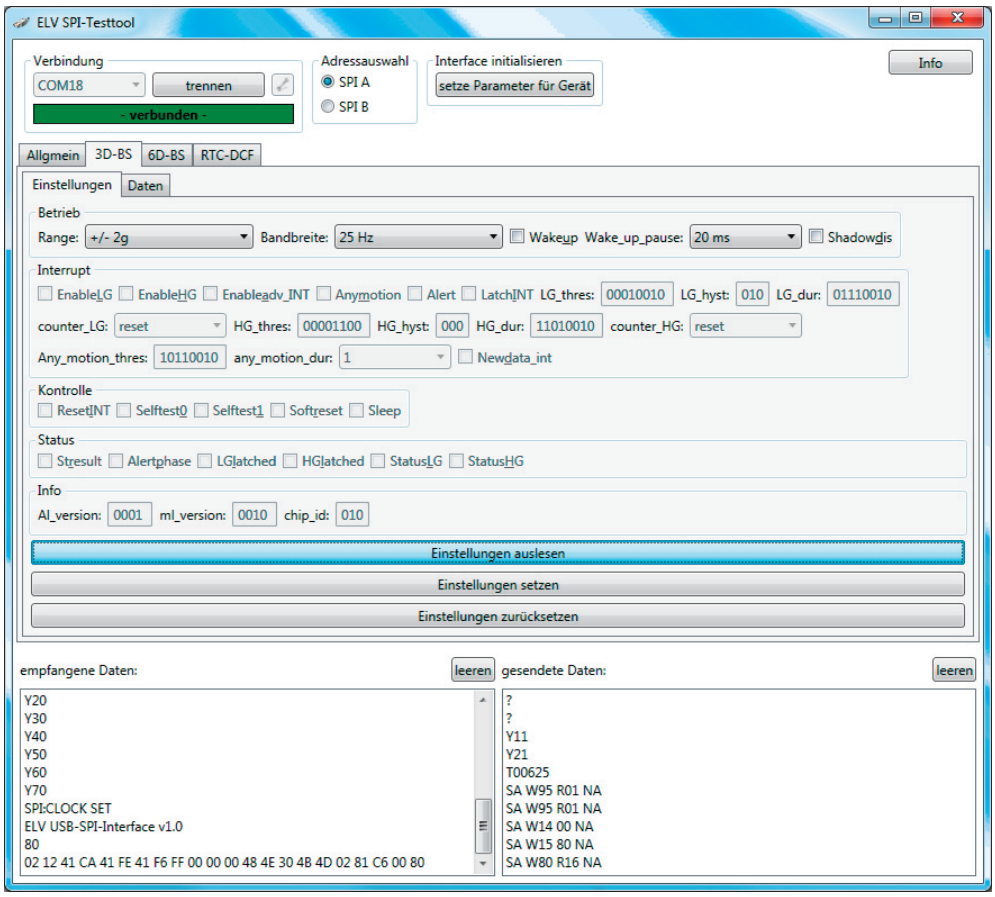


Bild 13: Konfiguration und Kommunikation des 3-Achsen-Beschleunigungssensors 3D-BS

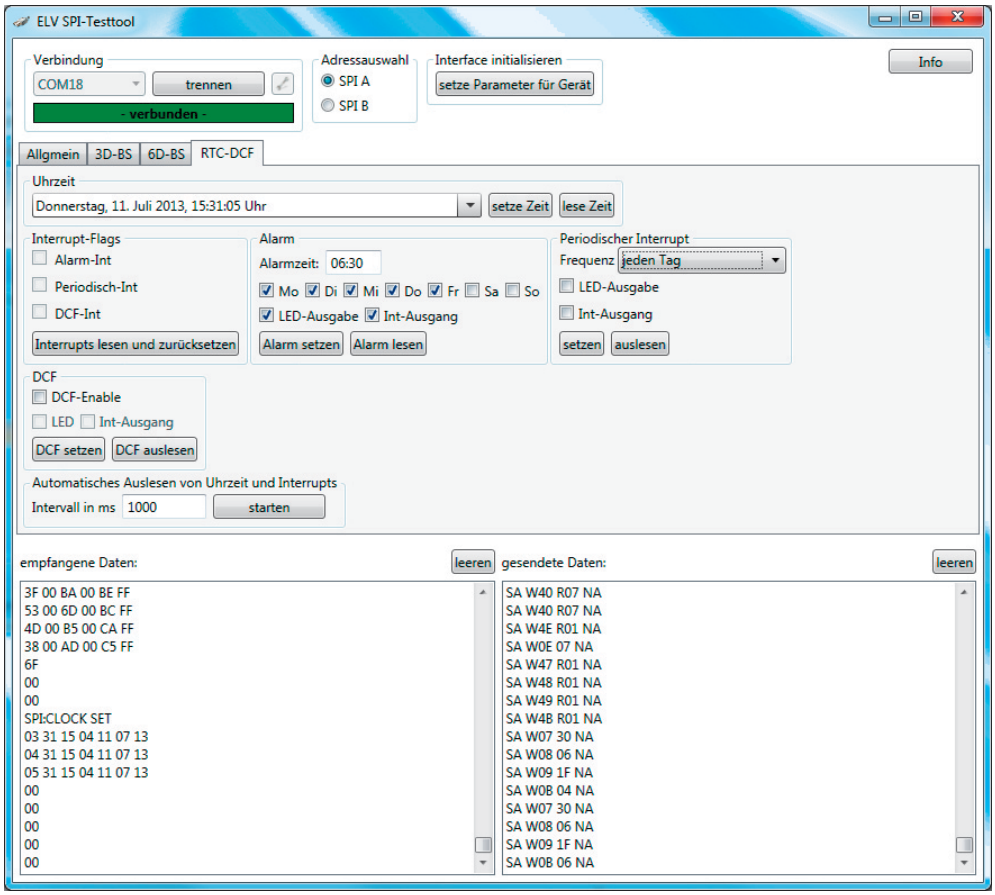


Bild 14: Konfiguration und Kommunikation des RTC-DCF-Bausteins

Weitere Infos:

[1]:www.elv.de:
Webcode: #1268
[2]:HTerm:
www.der-hammer.info/terminal/index.htm