

Neu! Texas-Instruments-Treiber IC TLC59116!

I2C-Bus-LED-Treiber

Steuern Sie 16 LEDs über nur 2 Leitungen

Diese universell einsetzbare Treiberschaltung verfügt über 16 LED-Ausgangskanäle, die einzeln über einen I2C-Bus steuerbar sind. Dabei ist die Helligkeit jeder LED über PWM einstellbar. Ebenso ist der LED-Strom in einem weiten Bereich per Software wählbar. Bis zu 14 dieser Treiberschaltungen können an einem I2C-Bus betrieben werden, so dass man hierüber auch größere LED-Anordnungen, etwa einen Sternenhimmel, individuell ansteuern und betreiben kann.

Komplex steuern über zwei Leitungen

Zur Steuerung von elektronischen Komponenten innerhalb einer Schaltung hat sich das in den 80er Jahren von Philips entwickelte serielle I2C-Bussystem bewährt. Der Vorteil dieses Systems liegt darin, dass nur zwei Signalleitungen benötigt werden und dass das Handling relativ einfach ist. Die Kommunikation findet in der Regel zwischen einem Master (Mikrocontroller) und einem Slave (Peripherie) statt. Mittlerweile sind fast alle Mikrocontroller-Peripheriebausteine – wie z. B. Speicher, Anzeigetreiber, ADC/DAC – mit einer I2C-Schnittstelle erhältlich. So gibt es auch für die Ansteuerung von LEDs zahlreiche Chips auf dem Markt. Einen relativ neuen und innovativen I2C-LED-Treiber von Texas Instruments, den TLC59116, wollen wir hier in einer praktischen Schaltung vorstellen. Das Besondere an diesem Chip ist, dass alle Parameter über den seriellen Bus einstellbar sind. Dies erfolgt über das direkte Ansprechen der Steuerregister des TLC59116. Jede einzelne der insgesamt 16 an den Treiber angeschlossenen LEDs kann separat in Schritten von 0 bis 255 (8 Bit) in der Helligkeit eingestellt werden.

Ein weiteres Feature sind die stromgeregelten, als Konstantstromsenke ausgeführten Ausgänge, die

es ermöglichen, LEDs ohne Vorwiderstand zu betreiben, wobei der LED-Strom in einem weiten Bereich per Software (Register) einstellbar ist (max. 20 mA).

Mittels steckbarer Jumper können auf der Platine 14 unterschiedliche Hardware-Adressen gewählt werden, so dass man max. 14 dieser Bausteine gleichzeitig an einem Bus betreiben kann.

Die Platine bietet neben dem direkten Bestücken mit LEDs die Möglichkeit, Subminiaturbuchsen für die LED-Ausgänge zu bestücken. LEDs in der Gehäuseform 1206 oder PLCC können auch direkt auf die Platine gelötet werden.

Für eine solche LED-Anordnung sind unzählige Anwendungen denkbar: von der einfachen Signal- oder Statusanzeige über komplexe Anordnungen wie einem „lebenden“ Sternenhimmel bis hin zur „bewegten“ Lichtskulptur aus Hunderten von LEDs, die alle einzeln ansteuerbar sind. Dabei kann die Ansteuerung je nach Bedarf von einem separaten Mikrocontroller,

Daten

Spannungsversorgung:	5–17 Vdc
Ausgänge:	16 Kanäle/max. 20 mA Konstantstrom
Interface:	Fast-I2C-Bus (bis 1 MHz)
I2C-Hardware-Adressen:	14 (über Jumper wählbar)
Sonstiges:	integrierte Temperatursicherung
Abm. (L x B):	70 x 39 mm

aber auch – über ein PC-Interface – von einem PC aus erfolgen. Beide Möglichkeiten werden wir hier betrachten.

Schaltung

Die Schaltung (Bild 1) besteht im Wesentlichen aus dem I2C-Baustein IC 1 vom Typ TLC59116 des Herstellers Texas Instruments (Bild 2 zeigt das Blockschaltbild). Dieser Baustein ist speziell für die Ansteuerung von bis zu 16 LEDs konzipiert.

Die Peripherie von IC 1 besteht aus nur wenigen Bauteilen. IC 1 benötigt eine stabile Spannung von 5 V, die mit dem Spannungsregler IC 2 aus der extern zugeführten Betriebsspannung (KL 1) gewonnen wird.

Mit dem Jumper J 1 erfolgt die Auswahl, ob die Schaltung über KL 1 (7–17 V) oder über die Anschlussleiste ST 1/ST 2 (5 V) versorgt wird. Die Dioden D 1 und D 2 dienen hier zur Entkopplung. Die Poly-Switches R 1 und R 2 sind als reversible (automatisch zurückstellende) Sicherung in einem Fehlerfall vorgesehen.

Mit den Jumpern J 4 bis J 7 wird die hardwaremäßige I2C-Adresse gewählt. Es können insgesamt 14 unterschiedliche Adressen eingestellt werden (siehe Installation). Mit den Widerständen R 3 und R 4 sowie den Jumpern J 2 und J 3 kann man für die Signalleitungen SDA und SCL bei Bedarf einen Pull-up realisieren. Im Normalfall sind die beiden Jumper nicht gebrückt. An die Ausgänge OUT 0 bis OUT 15 sind die 16 LEDs angeschlossen. Zur Unterdrückung von Störstrahlungen befinden sich in allen Signalleitungen zu den LEDs Chip-Ferrite. Soweit zur Schaltungsbeschreibung, kommen wir nun zur Anwendung selbst.

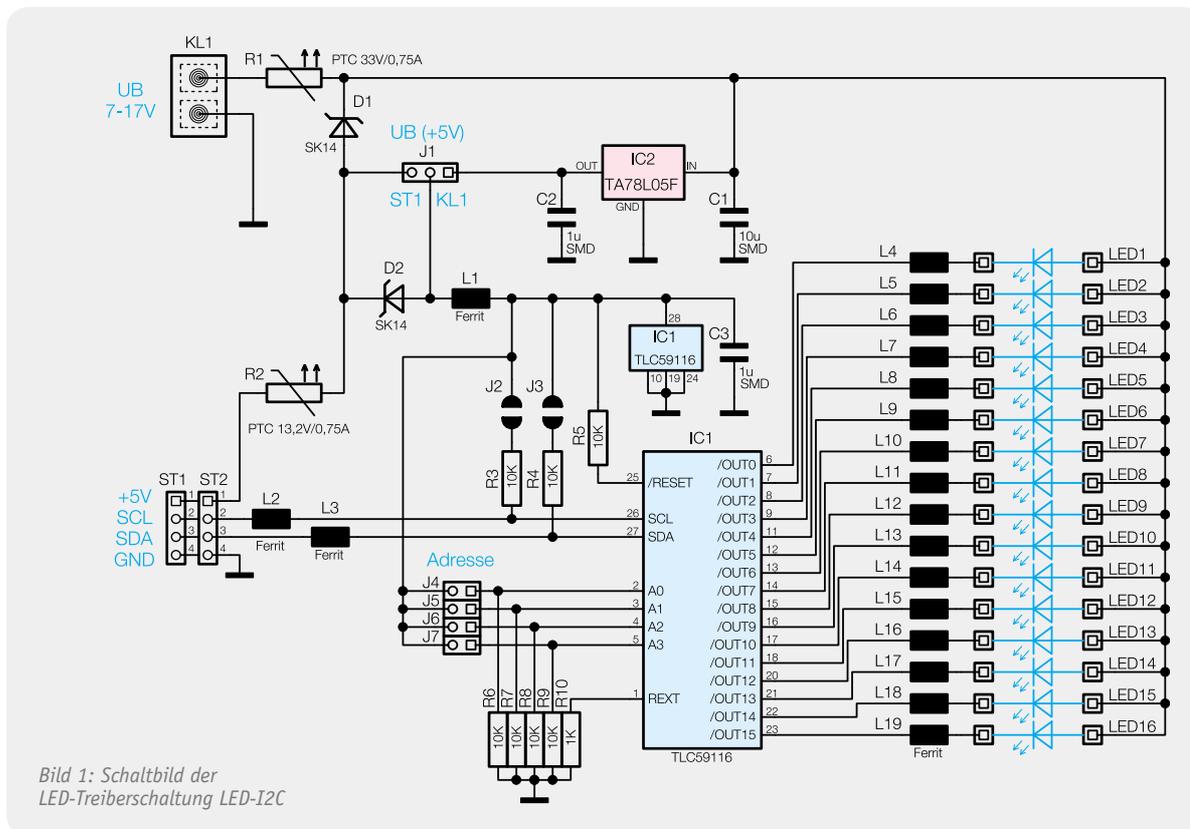


Bild 1: Schaltbild der LED-Treiberschaltung LED-I2C

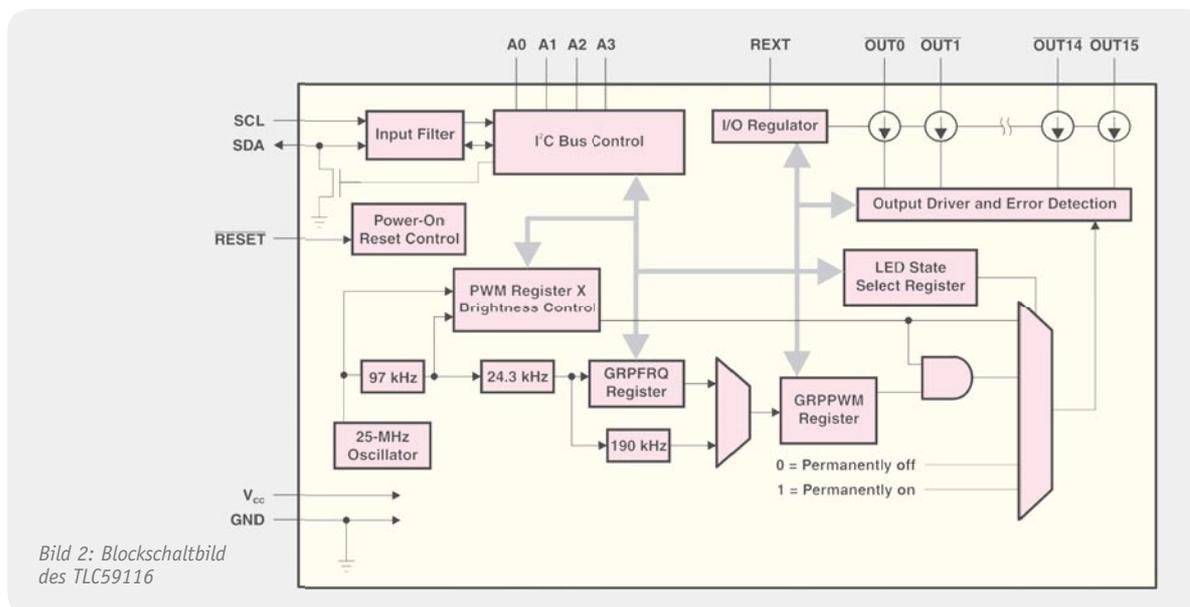


Bild 2: Blockschaltbild des TLC59116

Tabelle 1: Die Register des TLC59116

Register Nr.	Datenbits/Datenbytes								Name	Funktion	
Hex	D7	D6	D5	D4	D3	D2	D1	D0			
00	A12	A11	A10	SLEEP	SUB1	SUB2	SUB3	ALL CALL	MODE0	Controll-Register (genaue Beschreibung im Datenblatt)	
01	EFC LR	-	DMB LNK	-	OCH	-	-	-	MODE1		
02	00 bis FF								PWM 0	Helligkeit LED 1 (PWM 0-255)	
03	00 bis FF								PWM 1	Helligkeit LED 2 (PWM 0-255)	
04	00 bis FF								PWM 2	Helligkeit LED 3 (PWM 0-255)	
05	00 bis FF								PWM 3	Helligkeit LED 4 (PWM 0-255)	
06	00 bis FF								PWM 4	Helligkeit LED 5 (PWM 0-255)	
07	00 bis FF								PWM 5	Helligkeit LED 6 (PWM 0-255)	
08	00 bis FF								PWM 6	Helligkeit LED 7 (PWM 0-255)	
09	00 bis FF								PWM 7	Helligkeit LED 8 (PWM 0-255)	
0A	00 bis FF								PWM 8	Helligkeit LED 9 (PWM 0-255)	
0B	00 bis FF								PWM 9	Helligkeit LED 10 (PWM 0-255)	
0C	00 bis FF								PWM 10	Helligkeit LED 11 (PWM 0-255)	
0D	00 bis FF								PWM 11	Helligkeit LED 12 (PWM 0-255)	
0E	00 bis FF								PWM 12	Helligkeit LED 13 (PWM 0-255)	
0F	00 bis FF								PWM 13	Helligkeit LED 14 (PWM 0-255)	
10	00 bis FF								PWM 14	Helligkeit LED 14 (PWM 0-255)	
11	00 bis FF								PWM 15	Helligkeit LED 16 (PWM 0-255)	
12	00 bis FF								GRPPWM	Gruppen-PWM	
13	00 bis FF								GRPFREQ	Blinkfrequenz	
14	LDR3		LDR2		LDR1		LDR0		LEDOUT 0	LDRx = 00: LEDx Aus LDRx = 01: Ein LDRx = 10: Helligkeit wird von PWMx-Register bestimmt LDRx = 11: Helligkeit wird von PWMx- und GRPPWM-Register bestimmt	
15	LDR7		LDR6		LDR5		LDR4		LEDOUT 1		
16	LDR11		LDR10		LDR9		LDR8		LEDOUT 2		
17	LDR15		LDR14		LDR13		LDR11		LEDOUT 3		
18	02 bis FF								SUBADR1	I2C-Bus Subadresse 1	
19	02 bis FF								SUBADR2	I2C-Bus Subadresse 2	
1A	02 bis FF								SUBADR3	I2C-Bus Subadresse 3	
1B	00 bis FF								ALLCALLADR	LED All Call I2C bus address register	
1C	CM	HC	CC							IREF	CM = High/low current multiplier HC = Subcurrent CC = Current multiplier
1D	E8	E7	E6	E5	E4	E3	E2	E1	EFLAG1	jeder LED ist ein Errorflag zugeordnet Ex = 1: LED nicht angeschlossen	
1E	E16	E15	E14	E13	E12	E11	E10	E9	EFLAG2		

Die Register des TLC59116

Auf eine detaillierte Beschreibung des I2C-Busprotokolls verzichten wir an dieser Stelle, da dies den Rahmen des Artikels sprengen würde. Im Abschnitt „Elektronikwissen“ sind einige grundlegende Informationen zum Thema I2C-Bus zusammengefasst.

Eine sehr umfassende Beschreibung zum Thema I2C findet sich im ELVjournal-Artikel „USB-I2C-Adapter“ [1]. Der Text ist unter www.elvjournal.de online verfügbar.

In Tabelle 1 sind die wichtigsten Register des TLC59116 dargestellt, wobei das vollständige Studium des Datenblattes [4], wie bei allen Bausteinen dieser Art, natürlich für den Programmierer Pflicht ist. Um die Funktionen dieser Register zu testen, ist das ebenfalls unter [1] verfügbare USB-I2C-Testtool in Verbindung mit dem USB-I2C-Adapter sehr gut geeignet, wie wir im nächsten Abschnitt noch sehen werden.

Anwendungsbeispiele

Im Folgenden wollen wir einige Anschlussvarianten und Beispiele diskutieren.

Der Anschluss der LED-Treiberplatine an (vorhandene) eigene Mikrocontrollerschaltungen ist recht unproblematisch, da nur die Daten- und Masseleitungen zu verbinden sind. Hierbei ist ansonsten nur auf das richtige Stecken des Jumpers J 1 für die Betriebsspannung zu achten (siehe Abschnitt Inbetriebnahme).

Wer die Schaltung über einen PC steuern möchte, kann dies in Verbindung mit dem ELV-USB-I2C-Adapter realisieren, wie es in Bild 3 dargestellt ist.

Der USB-I2C-Adapter wird über einen seriellen COM-Port-Treiber angesprochen. Dies kann über ein Terminalprogramm wie Hyperterminal oder ein ähnliches Programm erfolgen. Zum Test aller Funktionen der LED-Treiberplatine steht unter www.elvjournal.de [1] das Programm „USB-I2C-Testtool“ zum Download bereit.

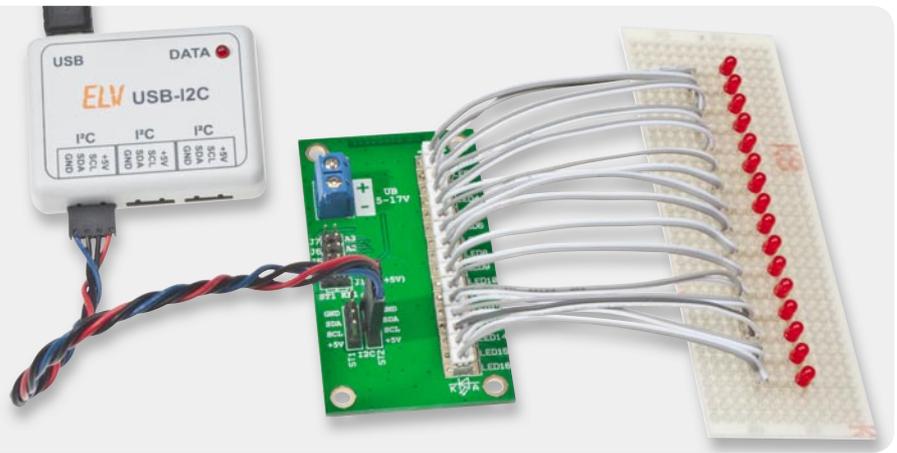


Bild 3: Anschluss des USB-I2C-Adapters an das LED-Board. Hier wurde ein abgesetztes LED-Board über Miniatur-Stiftbuchsen und die (hier gekürzten) passenden Anschlusskabel angeschlossen.

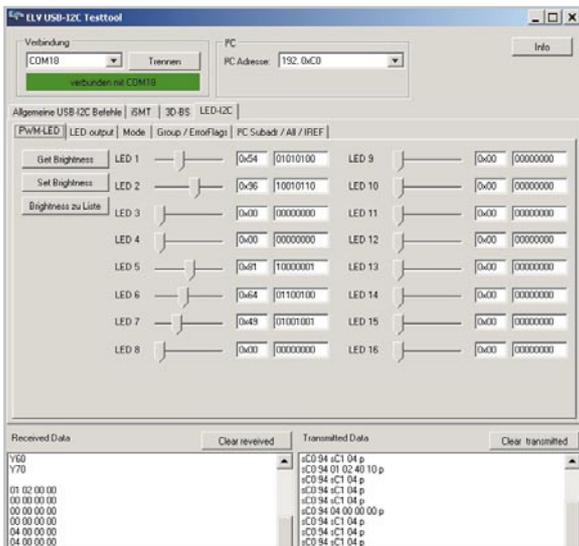


Bild 4: Das USB-I2C-Testtool in Aktion, hier die Einstellung der Helligkeit der einzelnen LEDs

In Verbindung mit dem USB-I2C-Adapter kann über dieses Programm jedes Register des TLC59116 angesprochen werden. Falls noch kein Treiber für den virtuellen COM-Port auf dem Rechner vorhanden ist, muss dieser zuerst installiert werden. Dieser Treiber der Firma Silabs steht ebenfalls unter [1] kostenlos zum Download bereit („Silabs Virtual-COM-Port-Treiber“). Ist dieser Treiber installiert, ist der USB-I2C-Adapter über den virtuellen COM-Port-Treiber wie ein serielles Gerät ansprechbar.

Nach dem Start des „USB-I2C-Testtools“ (siehe auch Screenshot Bild 4) ist zunächst der COM-Port im Feld „Verbindung“ einzustellen. Wie man den richtigen COM-Port ermittelt, ist auch in der Bedienungsanleitung zum USB-I2C-Adapter nachzulesen. Als Nächstes muss das entsprechende I2C-Gerät ausgewählt werden, da diese Software auch zum Testen weiterer ELV-I2C-Komponenten geeignet ist.

Anschließend können über entsprechende Reiter alle Register des TLC59116 ausgewählt und angesprochen werden. Es besteht auch die Möglichkeit, einzelne Befehlssequenzen in einem Editor-Fenster zu speichern, um diese dann mit einem Terminalprogramm verwenden zu können. Hier noch ein Hinweis: Die LED-Ausgänge sind erst aktiv, wenn die LED-Output-Register gesetzt sind.

Ein weiteres Anwendungsbeispiel zeigt Bild 5 mit einem Arduino-Board. Dieses weit verbreitete Mikrocontrollerboard basiert in der aktuellen Version auf einem Atmel-Controller „AtMega 328“. Über die Arduino-Entwicklungsumgebung kann das Board direkt über die USB-Schnittstelle programmiert werden. Hervorstechend am Arduino-Projekt: Es ist ein Open-Source-Projekt, sämtliche Software wie Compiler, Libraries usw. ist kostenlos im Internet erhältlich. In [2] haben wir dazu ausführlich berichtet, einschließlich der Vorstellung eines kompletten Lernpaketes inkl. Handbuch.

Der Anschluss des I2C-LED-Treibers an das Arduino-Board erfolgt über 4 Anschlussleitungen:

LED-I2C	→	Arduino
+5 V	→	+5 V
GND	→	GND
SDA	→	Analog-in 3 (Stiftleiste Pin 4)
SCL	→	Analog-in 4 (Stiftleiste Pin 5)

Für die Inbetriebnahme steht unter www.elvjournal.de [3] ein Demoprogramm mit entsprechenden Ansteuererroutinen für das Arduino-Board zur Verfügung.

Nachbau und Inbetriebnahme

Die Platine wird bereits mit SMD-Bauteilen bestückt geliefert, so dass nur die mechanischen Bauteile (Steckverbinder und Jumper) bestückt werden müssen. Die Position dieser Bauteile ist im Bestückungsdruck bzw. im Platinenfoto dargestellt. Auf der Platinenunterseite sind die Bauteilanschlüsse zu verlöten. Vor dem Einsatz der Schaltung sollte die Platine noch einmal auf evtl. fehlerhafte Lötstellen und Lötzinnbrücken untersucht werden.

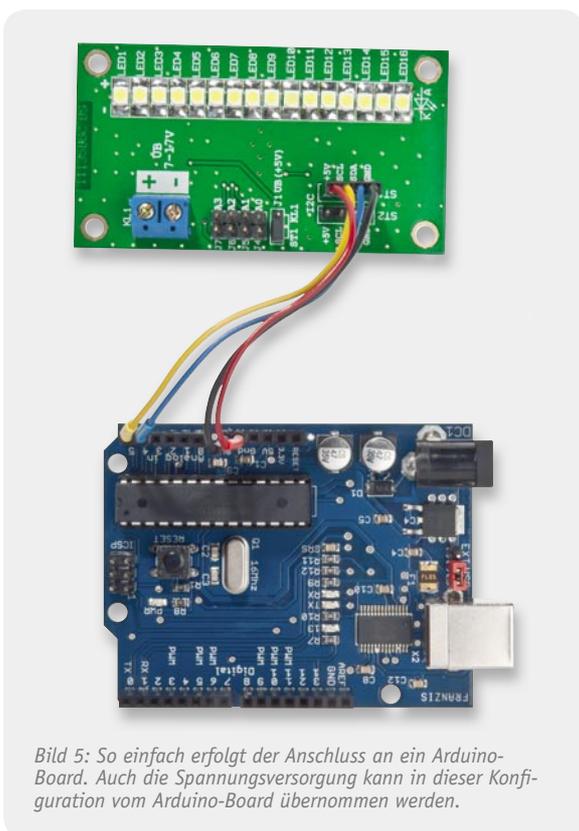


Bild 5: So einfach erfolgt der Anschluss an ein Arduino-Board. Auch die Spannungsversorgung kann in dieser Konfiguration vom Arduino-Board übernommen werden.

Tabelle 2

Konfiguration von Jumper 1

Stellung KL 1	Spannungsversorgung erfolgt über Klemme KL 1/7–17 V)/An ST 1 liegen ca. 4,7 V für die Versorgung des I2C-Masters an
Stellung ST 1	Spannungsversorgung erfolgt über Stiftleiste ST 1/ST 2 (5 V)

Nachdem die Schaltung fertig aufgebaut ist, kann die Inbetriebnahme erfolgen.

Die Spannungsversorgung kann auf zwei unterschiedliche Arten bewerkstelligt werden. Mit Jumper J 1 wird die Auswahl vorgenommen (Tabelle 2).

Im Regelfall erfolgt die Spannungsversorgung über die Klemme KL 1, wobei hier eine unstabilisierte Gleichspannung im Bereich von 7 bis 17 V verwendet werden kann. Der Jumper ist in Stellung „KL 1“ zu bringen. Möchte man die vom Spannungsregler stabilisierte Spannung auch für die Ansteuerschaltung nutzen, kann an der Stiftleiste ST 1/ST 2 diese 5-V-Spannung (max. 50 mA) entnommen werden.

Im zweiten Fall, wenn also Jumper J 1 auf „ST 1“ gebrückt ist, wird die Betriebsspannung über ST 1 (Pin 1/5 V) zugeführt. Hierbei ist zu beachten, dass diese stabilisiert sein muss. Auch für die Anzahl der LEDs gibt es eine Einschränkung – mit nur 5 V Betriebsspannung lässt sich pro LED-Ausgang jeweils nur eine LED ansteuern. Bei zwei in Reihe geschalteten LEDs wäre unter Umständen die Gesamtflussspannung höher als die zugeführte Betriebsspannung. Beispiel: Zwei weiße LEDs mit einer Flussspannung von jeweils 3,3 V ergeben eine Gesamtflussspannung von 6,6 V.

Generell können an den Ausgängen beliebig viele LEDs in Reihe geschaltet werden, solange die Betriebsspannung ca. 2 V höher ist als die Gesamtflussspannung der angeschlossenen LEDs. In diesem Fall ist die Spannungsversorgung über KL 1 zwingend erforderlich.

Kommen wir nun zum Anschluss der LEDs. Die Anschlusspunkte für die LEDs sind universell ausgelegt (siehe Bild 6). Im einfachsten Fall werden die LEDs direkt auf die Platine gelötet (siehe Bild 7), was allerdings nur bei der Bauform PLCC oder 1206 möglich ist. Will man die LEDs abgesetzt von der Platine betreiben, so lässt sich deren Anschluss sehr elegant durch optional zum Bausatz erhältliche Stiftbuchsen und passende Anschlusskabel (Bild 8) realisieren.

Widerstände:

1 k Ω /SMD/0603	R10
10 k Ω /SMD/0603	R3–R9
Polyswitch, 13,2 V, 0,75 A, SMD, 1812	R2
Polyswitch, 33 V, 0,75 A, SMD, 1812	R1

Kondensatoren:

1 μ F/SMD/1206	C2, C3
10 μ F/25 V/SMD/1210	C1

Halbleiter:

TLC59116IPWR/SMD/TI	IC1
TA78L05F/SMD/Toshiba	IC2
SK14/SMD	D1, D2

Sonstiges:

Chip-Ferrite, 0603, 600 Ω bei 100 MHz, 200 mA	L1–L19
Schraubklemmleiste, 2-polig, print	KL1
Stiftleisten, 1x 4-polig, gerade, print	ST1, ST2
Stiftleiste, 1x 3-polig, gerade, print	J1
Stiftleiste, 2x 4-polig, gerade, print	J4–J7
Jumper	J1, J4–J7

Stückliste

Wie man die Schaltung an eine Mikrocontrollerschaltung anbindet, ist im Abschnitt „Anwendungsbeispiele“ dargestellt. In einigen Anwendungen kann es erforderlich sein, die Signalleitungen mit einem zusätzlichen Pull-up-Widerstand zu versehen. Hierzu sind die Jumper J 2 und J 3 (Lötbrücken) zu schließen. Dies erfolgt durch Aufbringen einer Lötzinnebrücke auf die Flächen von J 2 und J 3. **ELV**

Weitere Infos:

- [1] ELVjournal-Artikel zum ELV-USB-I2C-Adapter:
Webcode: #1162
- [2] Das Arduino-Projekt, „ELVjournal“ 6/2010, S. 88
- [3] ELVjournal-Artikel zum LED-I2C-Board 98377:
Webcode: #1163
- [4] Datenblatt TLC59116, Texas Instruments:
Webcode: #1164

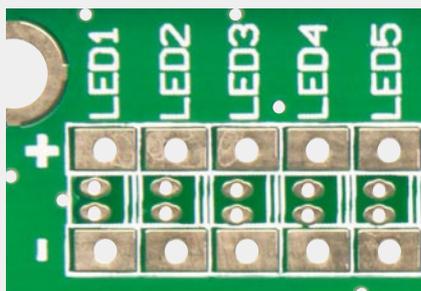


Bild 6: Die Anschlusspunkte für die LED-Ausgänge sind universell ausgeführt.

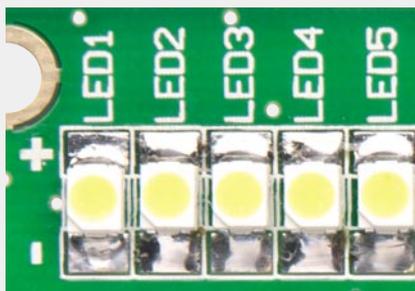


Bild 7: Bestückung mit SMD-LEDs, direkt auf die Platine gelötet

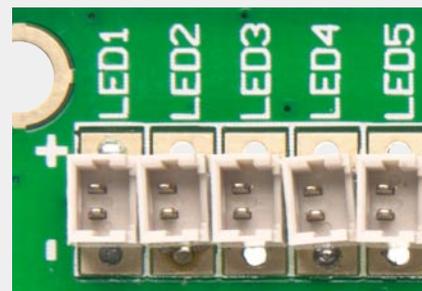
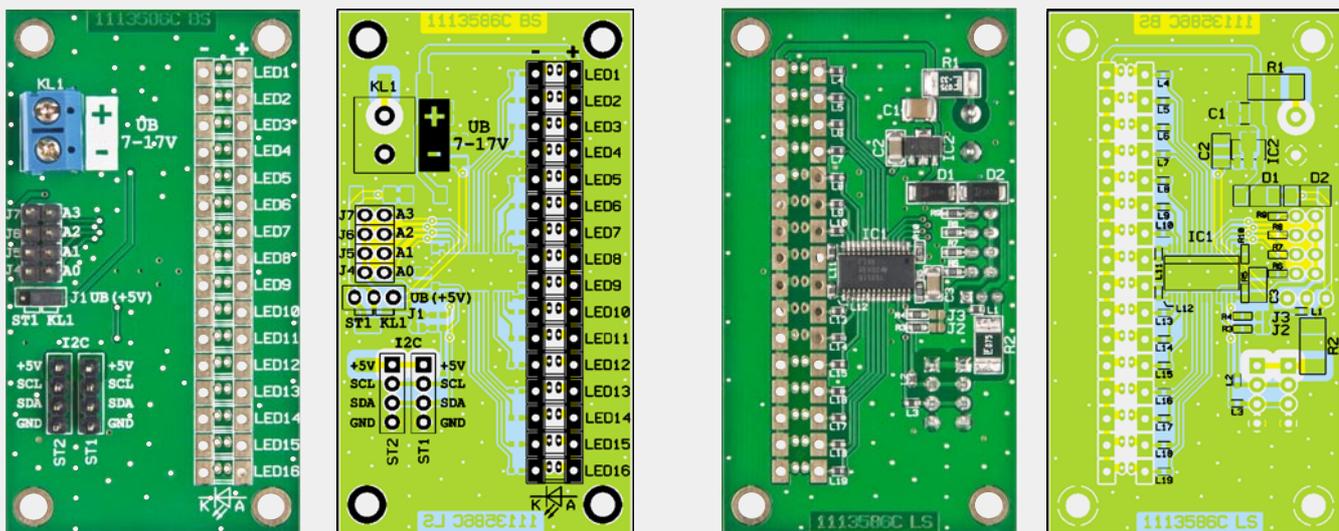
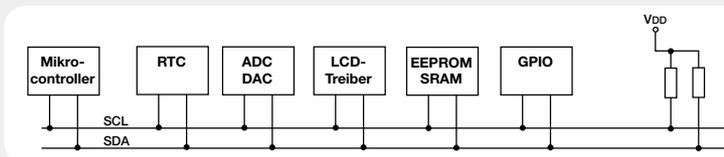


Bild 8: Alternative für abgesetzte LEDs: Miniatur-Steckbuchsen zur Aufnahme von konfektionierten Anschlusskabeln

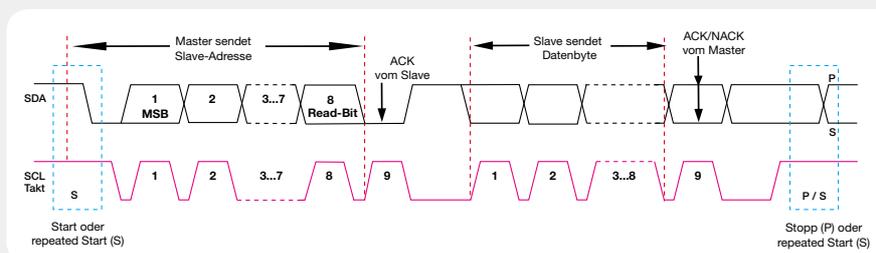


Ansicht der bestückten Platine des LED-I2C-Boards mit den zugehörigen Bestückungsplänen, links Oberseite, rechts Unterseite der Platine

Eine Beispielkonfiguration für verschiedene Komponenten am I2C-Bus



Ein kompletter Datenübertragungszyklus, bei dem der Master vom Slave 1 Datenbyte anfordert und erhält.



Der I2C-Bus

Die Philips Semiconductors Division stellte bereits in den 80er Jahren den seriellen Datenbus I2C (Inter-Integrated Circuit Bus) zur aufwandsarmen Steuerung von Komponenten der Heimelektronik vor. Später wurde I2C zum Industriestandard, aus lizenzrechtlichen Gründen findet man auch häufig die Bezeichnung „TWI“ (Two Wire Interface) hierfür, z. B. bei den verbreiteten AVR-Controllern.

Der I2C-Bus ist ein synchroner, serieller Zweidraht-Bus, auf dem mindestens ein Master-Gerät und die adressierten Slave-Geräte mittels eines in Hard- und Software realisierten Protokolls miteinander kommunizieren.

Dies erfolgt, je nach Bus-Spezifikation, mit einer maximalen Taktrate von 3,4 MHz, üblich sind auch die Taktraten 100 und 400 kHz. Vorgegeben wird die maximale Taktrate durch den langsamsten Baustein am Bus. Die Buslänge darf bei einer maximalen Buskapazität von 400 pF (je Bus-Segment, über Expander erweiterbar) je nach Geschwindigkeit bis zu mehreren Metern betragen. Der Bus besteht aus zwei genutzten Leitungen, der Taktleitung SCL (Serial Clock Line) und der Datenleitung SDA (Serial Data Line), die die beteiligten Geräte miteinander verbinden.

Der als Master fungierende Baustein ist für den definierten Ablauf der Kommunikation zuständig, also Steuerung der Abläufe, Generierung des Taktsignals und Adressierung sowie Datenversand/

-empfang. Der Slave empfängt Daten und bestätigt deren Empfang (Acknowledge). Natürlich kann auf Anforderung (Master setzt das Read-Bit in der Adressierung) auch der Slave Daten an den Master senden.

Eine komplette Datenübertragung erfolgt byteweise und beginnt mit der Adresse des angesprochenen Slave-Gerätes. Das höchstwertigste Bit eines Bytes wird dabei zuerst übertragen (MSB first). Danach quittiert der Slave den Erhalt mit dem Acknowledge-Signal (ACK), indem er, während das nächste Taktsignal vom Master kommt, die SDA-Leitung auf „low“ zieht. Er signalisiert damit auch, dass er bereit ist, ein weiteres Byte zu empfangen. Der Master kann nach der Datenübertragung entweder eine Stopp-Bedingung zum Abschluss des Datentransfers generieren oder eine sogenannte Repeated-Start-Bedingung für einen neuen Datentransfer.

Eine sehr ausführliche Beschreibung des kompletten Datentransfers und der zugehörigen Bedingungen findet sich in der Dokumentation, die unter [1] verfügbar ist.