Arduino-Shield

I²C-Schnittstelle

Inkl. I²C-Temperatur- und 4-Tasten-Modul

Vielseitige I²C-7-Segment-Anzeige – für Arduino oder solo

Ein vierstelliges Display über nur drei Adern ansteuern – das ist ein echter Problemlöser für viele Fernanzeige-Aufgaben. Dieses Display ist mit vier weißen 7-Segment-LED-Anzeigen ausgestattet und kann über den I²C-Bus angesteuert werden. Die Platine ist universell einsetzbar, z. B. auch als Arduino-Shield. Als erste Anwendungen gibt es noch einen zusätzlichen I²C-Temperaturchip sowie 4 Taster, die praktische Lösungen wie z. B. eine Stoppuhr oder eine Temperaturanzeige erlauben.

Technische Daten

I2C-4DLED (SAA1064)	
Kurzbezeichnung:	I2C-4DLED
Versorgungsspannung:	5 VDC
ULED	5,5–15 VDC
Stromaufnahme:	ca. 10 mA (ohne aktive LEDs)
	170 mA max. (alle Segmente ein/max. LED-Strom)
Interface:	I²C-Bus
Sonstiges:	LED-Strom einstellbar, 3 bis 21 mA in 3-mA-Schritten
Anzeige:	4x 7-Segment (weiß)
Temperaturbereich:	-10 bis +55 °C
Abmessungen (gesamte Platine):	55 x 70 mm
Gewicht:	33 g
Temperaturbaustein MCP9801	
Versorgungsspannung:	2,7–5,5 VDC
Stromaufnahme:	400 µA
Auflösung:	9 bis 12 Bit
Messbereich:	-10 bis +85 °C
Interface:	I²C-Bus
Genauigkeit:	±1 °C (-10 bei +85 °C), ±0,5 °C (bei 25 °C)
Abmessungen (Platine):	12 x 15 mm
Gewicht:	1 g

Vielseitig einsetzbar

Ein Anzeigebaustein wie der hier vorgestellte ist tatsächlich sehr vielseitig einsetzbar, sowohl als abgesetzte Stand-alone-Anzeige, die über den Bus angesteuert wird, als auch als direkt auf einen Arduino aufsteckbare Anzeige für verschiedenste Aufgaben. Auch hier ist die Anzeige über die nur zwei Busleitungen praktisch, spart man doch viele der immer zu knappen Controller-Ports.

Die Platine des Anzeigemoduls besteht eigentlich aus drei Teilplatinen: der Anzeigeplatine selbst, einer Platine für einen I²C-Temperatursensor und einer Platine mit vier Tastern. Der Einsatz ist als Ganzes ebenso möglich wie in Form von drei Einzelplatinen. Ein individueller Einsatzfall wäre etwa eine kleine, von einem

AVR gesteuerte Uhr-/Thermometer-Lösung. Die mechanische Ausführung als Arduino-Shield liegt nahe, denn kaum eine Mikrocontroller-Plattform ist derzeit beliebter als diese einfach beherrschbare Plattform mit der „anhängenden" einfachen Entwicklungssoftware. Und schließlich kann man das Display auch über einen Protokoll-Umsetzer als abgesetzte Anzeige an einem PC-System betreiben.

Warum die LED-Anzeige? Durch den Einsatz von modernen, sehr hellen weißen 7-Segment-Anzeigen in Verbindung mit einer getönten Frontplatte aus speziellem Kunststoff wird eine sehr gute Ablesbarkeit erzielt. Und schließlich kann man mit verschiedenen Farbfilterfolien eine völlig individuelle Anzeigefarbe realisieren – ein großer Vorteil der weißen Anzeigen. Für die verschiedenen Einsatzvarianten stehen zwei unterschiedliche Frontplattengrößen zur Verfügung.

Die Basis – Displaycontroller SAA1064

Die Basis des Anzeigebausteins bildet der I²C-Displaycontroller SAA1064. Diesen Baustein gibt es bereits seit dem Jahr 1991, er wird von der Fa. NXP (früher Philips) hergestellt. Nicht zuletzt wegen seiner einfachen Programmierung über den I²C-Bus erfreut sich der SAA1064 seit Jahrzehnten großer Beliebtheit. Im Multiplexbetrieb stehen insgesamt 32 LED-Ausgänge zur Verfügung. Somit können vier 7-Segment-Anzeigen betrieben werden. Der Begriff „7 Segment" ist eigentlich nicht richtig, denn bei fast allen diesen Anzeigen findet sich auch noch ein zusätzlicher Dezimalpunkt (DP), was die Gesamtzahl der Segmente auf 8 erhöht. Im Prinzip könnten die Ausgänge auch mit einzelnen LEDs beschaltet werden, denn jedes Segment der Anzeige ist einzeln ansteuerbar.

Die Programmierung ist recht einfach. Möchte man ein Segment zum Leuchten bringen, wird einfach eine 1 in das korrespondierende Bit des entsprechenden „Data Digit Register" geschrieben. Zudem kann die Helligkeit in 8 Stufen über den LED-Strom eingestellt werden. Die detaillierten Informationen zur Programmierung finden sich im Datenblatt [1]. In Tabelle 1 werden die einzelnen Register beschrieben.

Tabelle 1

Resister des SAA1064

Nr.	MSB							LSB	Beschreibung
	0	1	1	1	0	A1	A0	0	Slave-Adresse
	0	0	0	0	0	SC	SB	SA	Instruction-Byte
00	X	C6	C5	C4	C3	C2	C1	C0	Control-Byte
01	D17	D16	D15	D14	D13	D12	D11	D10	Data-Digit 1
02	D27	D26	D25	D24	D23	D22	D21	D20	Data-Digit 2
03	D37	D36	D35	D34	D33	D32	D31	D30	Data-Digit 3
04	D47	D46	D45	D44	D43	D42	D41	D40	Data-Digit 4

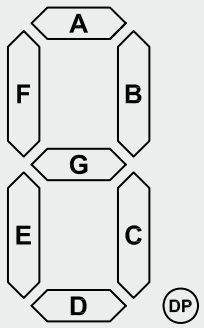
Tabelle 2

Resister des Control-Bytes

Bit	Beschreibung
C0 = 0	„static mode" – es werden nur Digit 1 und Digit 2 angezeigt
C0 = 1	„dynamic mode" – Digit 1 bis Digit 4 werden angezeigt
C1 = 0/1	Digit 1 + 3 sind ein- bzw. ausgeschaltet
C2 = 0/1	Digit 2 + 4 sind ein- bzw. ausgeschaltet
C3 = 1	alle Segmente eingeschaltet (dient zum Displaytest)
C4 = 1	addiert 3 mA zum LED-Ausgangsstrom
C5 = 1	addiert 6 mA zum LED-Ausgangsstrom
C6 = 1	addiert 12 mA zum LED-Ausgangsstrom

Tabelle 3

Segment	Register			
	Digit 1	Digit 2	Digit 3	Digit 4
A	D10	D20	D30	D40
B	D11	D21	D31	D41
C	D12	D22	D32	D42
D	D13	D23	D33	D43
E	D14	D24	D34	D44
F	D15	D25	D35	D45
G	D16	D26	D36	D46
DP	D17	D27	D37	D47



Slave-Adresse

Mit den Bits A1 und A0 kann die Slave-Adresse gewählt werden. Die hardwareseitige Einstellung erfolgt über die Jumper J7 bis J10 (Default 70 h = Jumper J7 geschlossen, siehe Schaltbild Bild 1). Hinweis: Es darf immer nur einer dieser Jumper geschlossen sein. Die Zuordnung der Adressen ist im Schaltbild dargestellt. Es sind vier verschiedene Slave-Adressen einstellbar, so dass auch vier SAA1064 gleichzeitig an einem Bus betrieben werden können.

Instruction-Byte

Mit den drei Bytes SC bis SA wird festgelegt, wo das folgende Byte geschrieben werden soll, z. B. 011 entspricht Register Nr. 3 (Data Digit 3). Hierdurch kann das Schreibprotokoll in der Länge verkürzt werden und es erfolgt ein schneller Zugriff auf bestimmte Register.

Control-Byte

Dieses Register steuert die verschiedenen Anzeigeoptionen. In Tabelle 2 sind die einzelnen Bits erklärt. Mit C4 bis C6 lässt sich der LED-Strom und somit die Helligkeit einstellen. Durch Kombination der drei Bits können folgende Ströme eingestellt werden: 0 mA/ 3 mA/6 mA/9 mA/12 mA/15 mA/18 mA/21 mA.

Data-Digit 1 bis 4

In diesen vier Registern (für jedes Digit ein Register) werden die einzelnen Segmente der Anzeige aktiviert. Wird das entsprechende Bit gesetzt, leuchtet das korrespondierende Segment auf. In Tabelle 3 ist die Zuordnung der Segmente zu den Registerbits dargestellt.

Der Temperaturbaustein MCP9801

Dieser Baustein verfügt über einen integrierten Temperatursensor, der mittels I²C-Bus ausgelesen werden kann. Vorteil hierbei ist, dass der Sensor nicht abgeglichen werden muss. Dies hat allerdings den kleinen Nachteil, dass die Genauigkeit „nur" bei ±1 °C liegt. Im Temperaturbereich um 25 °C ist die Genauigkeit etwas besser und liegt bei ±0,5 °C. Es gibt jedoch zahlreiche Anwendungen, bei denen diese Genauigkeit vollkommen ausreichend ist, wie z. B. Lüftersteuerungen und sonstige Temperaturregelschaltungen. Der

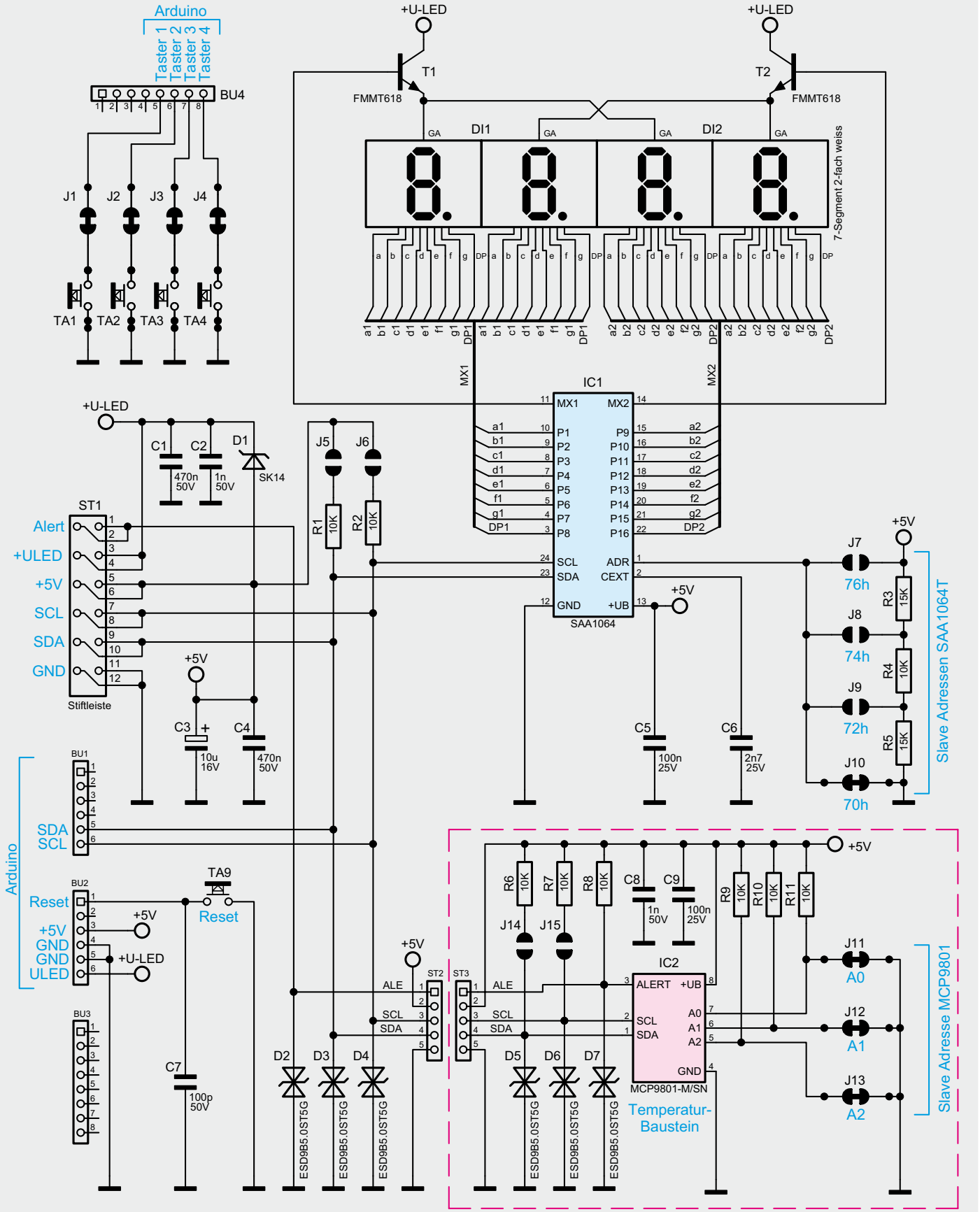


Bild 1: Das komplette Schaltbild der I²C-Anzeigeplatine inklusive Temperaturmessmodul

integrierte „Sigma Delta Analog to Digital Converter“ stellt eine Auflösung von 9 bis 12 Bit zur Verfügung. Unter den am Markt befindlichen integrierten I²C-Sensoren gehört der MCP9801 zur Oberklasse, was die Genauigkeit betrifft.

Der MCP9801 verfügt über einen Alarmausgang, der bei einer programmierbaren Temperatur aktiviert wird (Low-Pegel). Dies kann z. B. für einfache Thermostate genutzt werden. In Bild 2 ist die Sensorplatine dargestellt.

Die Slave-Adresse kann mit den Jumpern J11 bis J13 eingestellt werden. Defaultmäßig sind alle Jumper gebrückt, was der Adresse (90 h [schreiben] und 91 h [lesen]) entspricht. Tabelle 4 zeigt alle möglichen Slave-Adressen.

Für Demozwecke ist die Platine mit der Hauptplatine verbunden und somit auch gleichzeitig die Verbindung über den I²C-Bus hergestellt. Für praktische Anwendungen sollte die Sensorplatine abgesetzt mittels Flachbandkabel verbunden werden (siehe „Inbetriebnahme“).

Wie beim SAA1064 auch, kann hier nicht im Einzelnen auf die Programmierung mittels I²C-Bus eingegangen werden, da dies den Rahmen des Artikels sprengen würde. Unter [2] kann das Datenblatt vom Hersteller heruntergeladen werden. Als Hilfestellung bei der Programmierung sind auf der Produktseite im ELV-Web-Shop [3] natürlich Programmbeispiele für den Arduino zu finden.

Mit diesen Beispielen können dann eine einfache Stoppuhr und eine Temperaturmessung realisiert werden. So kann auch die korrekte Funktion getestet werden.

Bei der Stoppuhr kann mit Taste TA1 die Zeitmessung gestartet und mit Taste TA2 wieder gestoppt werden. Die angezeigten Zeiten sind nicht sehr genau (1 Sekunde Auflösung), sie dienen jedoch lediglich zu Demo- und Lernzwecken.

Beim Beispiel Temperaturmessung wird die Temperatur des MCP9801 ausgelesen und auf dem Display zur Anzeige gebracht. Mit den Tastern kann zwischen Grad Celsius (TA1), Kelvin (TA2) und Fahrenheit (TA3) gewählt werden.

Schaltung

Das Schaltbild des I²C-Displaymoduls ist in Bild 1 dargestellt. Die vielen Steckverbinder mögen zunächst etwas irritieren, sind aber für die Kontaktierung mit dem Arduino-Board notwendig.

Im oberen Teil des Schaltbildes sind die 7-Segment-Anzeigen mit dem Treiberbaustein SAA1064 zu erkennen. Wie schon erwähnt, werden die Anzeigen im Multiplexverfahren betrieben. Es sind jeweils immer nur zwei Anzeigen gleichzeitig aktiv. Die Steuerung geschieht mittels der beiden Transistoren T1 und T2. Der LED-Strom für die Anzeigensegmente wird durch interne Stromsenken begrenzt bzw. auf einen bestimmten Wert eingestellt (siehe Abschnitt „SAA1064“). Mit den Jumpern J7 bis J10 wird die Slave-Adresse eingestellt. Um die Zahl der Anschlüsse (Pins) am SAA1064 möglichst gering zu halten, erfolgt die Adresseinstellung nur über einen Anschluss (Pin1/ADR). Hier wird anhand der Spannung an diesem Pin die Adresse ausgewählt. Die Widerstände R3 bis R5 bilden einen Spannungsteiler, dessen Spannungsabgriffe mit Hilfe der Jumper J7 bis J10 auf den ADR-Pin geführt werden. Hier ist zu beachten, dass immer nur ein Jumper gebrückt sein darf.

Mit den beiden Jumpern J5 und J6 sind Pull-up-Widerstände für den I²C-Bus aktivierbar. Dies ist nur dann der Fall, wenn der Master (Host) keine Pull-up-Widerstände zur Verfügung stellt.

Für die Spannungsversorgung stehen zwei Eingänge zur Verfügung: +5 V und ULED.

Der zusätzliche Spannungseingang ULED kann zur Versorgung der LED-Anzeigen genutzt werden, wenn der maximale Strom über den Anschluss „+5 V“ nicht ausreicht. Über die Diode D1 ist der Eingang ULED entkoppelt (siehe auch Abschnitt „Separate Spannungsversorgung der 7-Segment-Anzeigen“).

Im gestrichelten Feld des Schaltbildes ist der separate Temperaturbaustein MCP9801 dargestellt. Diese Platine kann mechanisch und elektrisch von der Hauptplatine getrennt werden. Die Slave-Adresse wird mittels Jumper (J11 bis J13) eingestellt. Auch hier kann man Pull-up-Widerstände für den I²C-Bus durch Brücken der Jumper J14 und J15 aktivieren. Der MCP9801 darf nur mit einer maximalen Betriebsspannung von 5,5 V betrieben werden (siehe „Technische Daten“). Der als ALE gekennzeichnete Anschluss ist der Alarmausgang, der im Alarmfall (Überschreitung einer programmierbaren Temperatur) auf Low-Potential geschaltet wird (Open Collector).

Die Dioden D2 bis D7 sind spezielle Schutzdioden, die zum Schutz vor Spannungsspitzen oder statischen Entladungen dienen. Der dargestellte Taster TA9 ist für

Tabelle 4

Slave-Adressen des MCP9801				
J13	**J12**	**J11**	**Schreiben**	**Lesen**
0	0	0	90 h	91 h
0	0	1	92 h	93 h
0	1	0	94 h	95 h
0	1	1	96 h	97 h
1	0	0	98 h	99 h
1	0	1	9A h	9B h
1	1	0	9C h	9D h
1	1	1	9E h	9F h

0 = Jumper geschlossen/1 = Jumper offen

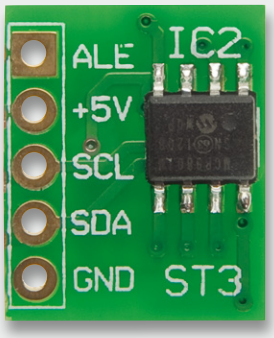


Bild 2: Die abtrennbare Platine mit dem I²C-Temperatursensor MCP9801

den Betrieb mit einem Arduino-Board gedacht, er erlaubt das Rücksetzen des Arduino-Hauptprozessors.

Nachbau

Der Nachbau ist durch die vorbestückten SMD-Bauteile einfach und ist recht schnell zu erledigen.

Zunächst ist eine Kontrolle der Bestückung entsprechend Bild 3 vorzunehmen. Dann folgt das Bestücken der bedrahteten Bauteile.

Die Bauteile werden von oben in die dafür vorgesehenen Bohrungen in der Platine eingesetzt und ihre Anschlüsse auf der Platinenunterseite verlötet. Wir beginnen die Bestückungsarbeiten mit dem Einsetzen des Elkos C3. Dieser Kondensator (Elko) wird liegend montiert, wie im Platinenfoto (siehe Bild 3) zu sehen. Beim Einsetzen des Bauteils ist unbedingt auf die richtige Polung zu achten. Der Pluspol ist durch den etwas längeren Anschlussdraht erkennbar. Falls die Anschlussdrähte schon vorkonfektioniert sind (Anschlussdrähte auf gleiche Länge gekürzt), ist die Polung auch am Aufdruck auf dem Gehäuse identifizierbar. In der Regel ist der Minuspol durch eine Strichmarkierung gekennzeichnet. Nun können die 7-Segment-Anzeigen eingesetzt und verlötet werden. Hierbei sollte man auf die korrekte Einbaulage achten, so dass sich der Dezimalpunkt (DP) auf der richtigen Seite befindet.

Der Einbau der Stiftleisten ist davon abhängig, wie man die Schaltung betreiben möchte (siehe „Inbetriebnahme“). Soll die Platine als Arduino-Shield eingesetzt werden, sind auf jeden Fall BU1 bis BU4 zu bestücken. Hier sei darauf hingewiesen, dass nicht benötigte Bedienelemente (Taster) auch nicht zwangsläufig bestückt werden müssen. Weitere Informationen sind im nächsten Abschnitt beschrieben. Je nach Anwendung stehen unterschiedliche Frontplatten zur Verfügung, die einen professionellen Einsatz z. B. durch Einbau in ein Gehäuse oder ein existierendes Gerät erlauben. In

Bild 4: Die beiden zur Verfügung stehenden Frontplattenversionen, oben mit Tastenfeld, unten ohne Tastenfeld

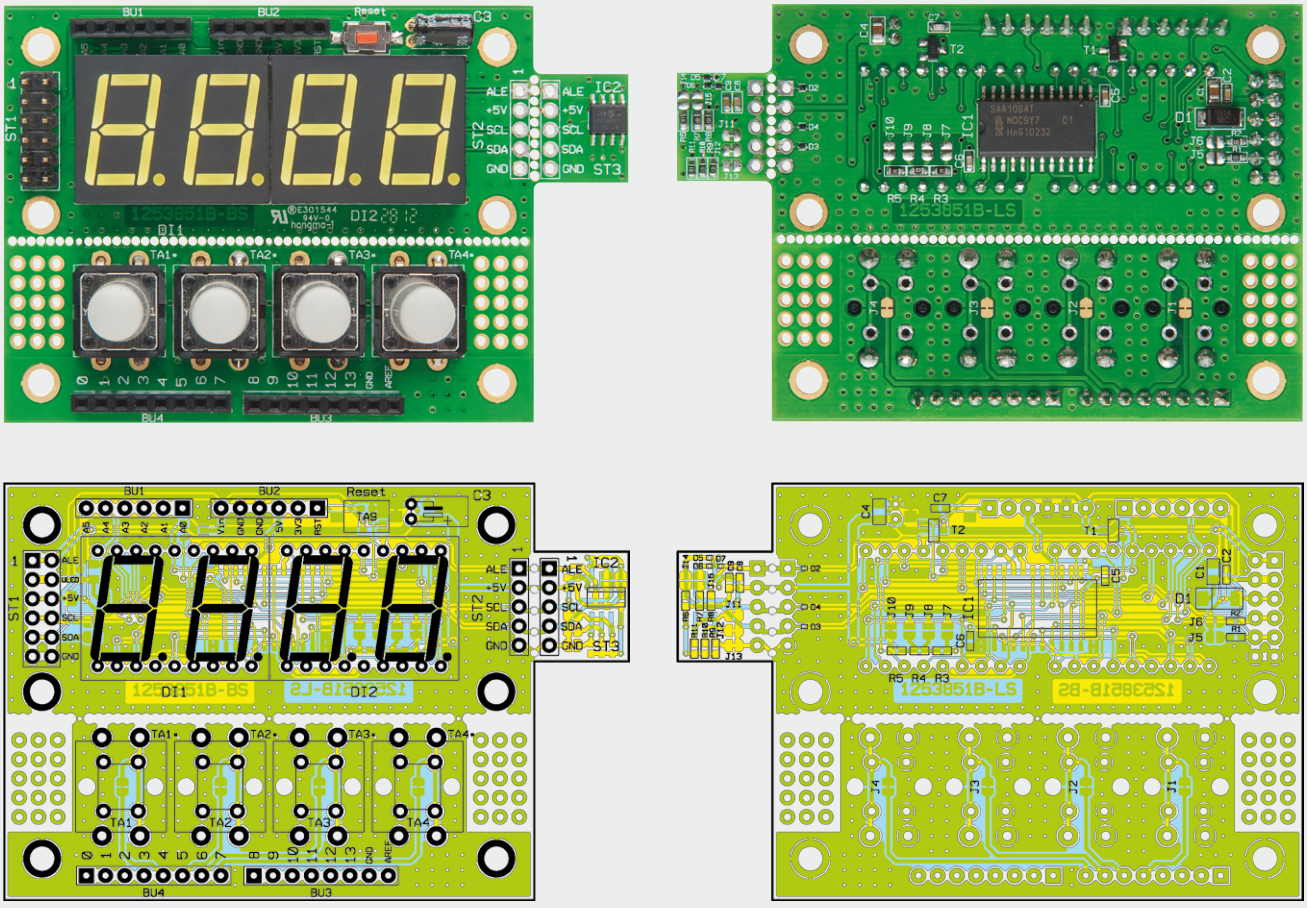


Bild 3: Die komplett bestückte Platine des Anzeigebausteins mit zugehörigem Bestückungsplan, links Oberseite, rechts Unterseite

Bild 4 sind beide Frontplattenversionen dargestellt.

Zur Befestigung der Platine dienen Abstandshalter, die ein 3-mm-Innengewinde aufweisen. Man kann somit von jeder Seite eine Schraube eindrehen. Bild 5 zeigt, wie diese Montage erfolgt. Auf der Frontplattenseite stehen zudem zwei verschiedene Schrauben (Bild 6) zur Verfügung. Je nach Geschmack kann man eine schwarze Senkkopfschraube, die mit der Frontplatte bündig abschließt, oder eine schwarze Innensechskantschraube verwenden.

Ein weiteres Feature des Bausatzes ist die farbliche Gestaltung der Anzeigen. Die Anzeigefarbe der LED-Anzeigen ist Weiß. Möchte man andere Darstellungsfarben, kann dies durch entsprechende Farbfolien realisiert werden. Bild 7 zeigt mehrere Beispiele dazu. Im optional erhältlichen Farbfolienset stehen sechs verschiedene Farbfilterfolien zur Verfügung. Die Folien werden rückseitig auf die Innenseite der Frontplatte geklebt. Es reicht, wenn man die Folie mit etwas durchsichtigem Klebeband (Tesafilm) an mehreren Stellen fixiert. Auf keinen Fall einen Cyanacrylat-Kleber (Sekundenkleber) verwenden, dieser erzeugt beim Abbinden einen Schleier, der nicht zu beseitigen ist (Materialoberfläche wird angelöst). Es gibt Modellbaukleber, die speziell für Acryl vorgesehen sind. Hiermit kann man, nach einem Test an verdeckter Stelle, die Folie ankleben.

Inbetriebnahme

Im folgenden Abschnitt werden die unterschiedlichen Betriebsarten beschrieben. Die Platine ist so ausgelegt, dass diese praktisch für alle möglichen Einsatzzwecke angepasst werden kann. Die Tasterplatine kann von der Anzeigeneinheit abgetrennt werden. Die Platine wird einfach durch Absägen entlang der Perforation abgetrennt.

Bild 5: So werden die Abstandshalter montiert.

Bild 6: Das komplette Montageset für die Frontplatte. Deren Befestigung erfolgt wahlweise mit den Senkkopfschrauben oder den Innensechskantschrauben.

Bild 7: Mit verschiedenen Farbfolien lässt sich eine individuelle Anzeigefarbe einstellen.

Stückliste

Widerstände:	
10 kΩ/SMD/0603	R1, R2, R4, R6–R8, R9–R11
15 kΩ/1 %/SMD/0603	R3, R5
Kondensatoren:	
100 pF/SMD/0603	C7
1 nF/SMD/0603	C2, C8
2,7 nF/SMD/0603	C6
100 nF/SMD/0603	C5, C9
470 nF/50 V/SMD/0805	C1, C4
10 µF/16 V	C3
Halbleiter:	
SAA1064T	IC1
MCP9801-M/SN/SMD	IC2
FMMT618/SMD	T1, T2
SK14/SMD	D1
ESD9B5.0ST5G/SMD	D2–D7
7-Segment-Doppelanzeige/weiß	DI1, DI2
Sonstiges:	
Buchsenleisten, 1x 6-polig, gerade, print	BU1, BU2
Buchsenleisten, 1x 8-polig, gerade, print	BU3, BU4
Miniatur-Drucktaster, 1x ein, SMD	TA9
Mini-Drucktaster, B3F-4050, 1x ein	TA1–TA4
Tastkappen, 10 mm, grau	TA1–TA4
Stiftleiste, 2x 6-polig, gerade, print	ST1
50 cm Flachbandkabel, AWG28, 5-polig, grau	

Die Temperatursensorplatine (siehe Bild 2) ist ebenfalls durch eine Perforation mit der Hauptplatine mechanisch und elektrisch verbunden.

Bei der abgesetzten Montage der Temperatureinheit müssen die Jumper J4, J5, J14 und J15 unbedingt gebrückt (verlötet) werden! Es muss sich ein Gesamtwiderstandswert von ca. 5 kΩ für die Pull-up-Widerstände ergeben. Nur so ist ein störungssicherer Betrieb gewährleistet.

Verbleibt die Temperatursensorplatine an der Hauptplatine, gibt es eine Besonderheit zu beachten. Durch die leichte Erwärmung der 7-Segment-Anzeigen kommt es zu einer Temperaturerhöhung auf der Platine und im Umfeld um mehrere Grad Celsius. Für Demozwecke kann die Temperatursensorplatine an der Hauptplatine verbleiben. Für praktische und reale Temperaturmessungen ist die Platine aber unbedingt abgesetzt zu montieren, wie in Bild 8 zu sehen. Die Kabellänge sollte nicht mehr als 50 cm betragen. Nur in dieser Betriebsart findet keine Beeinflussung durch Wärme produzierende Bauteile wie z. B. die Anzeigen statt.

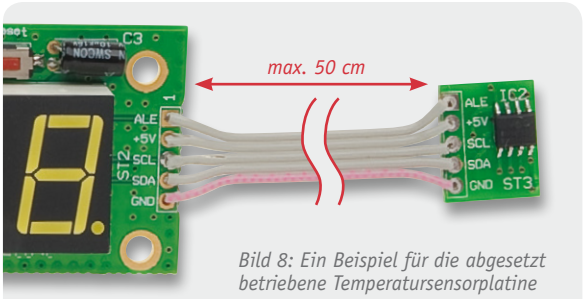


Bild 8: Ein Beispiel für die abgesetzt betriebene Temperatursensorplatine

Die Sensorplatine kann natürlich auch für andere (eigene) Anwendungen, die über den I²C-Bus kommunizieren, eingesetzt werden. In Bild 9 sind alle verfügbaren Anschlüsse dargestellt, die auf die Buchsenleisten BU1 bis BU4 nach außen geführt sind.

An Stiftleiste ST1 sind mit Ausnahme der Taster-kontakte ebenfalls alle Anschlüsse zugänglich. Die Bezeichnung ist hier auf der Platine aufgedruckt. Die Stiftleiste ST1 bietet die Möglichkeit, eine Standard-stiftleiste 2x 6 zu bestücken, auf die eine passende Buchsenleiste aufgesteckt werden kann. Die Bestückung der Stiftleiste kann auch auf der Rückseite erfolgen, wenn die Platine mit einer Frontplatte betrieben wird!

Für einen ausreichenden Schutz vor elektrostatischen Entladungen ist der Einbau in ein geeignetes Gehäuse erforderlich, damit die Schaltung nicht durch eine Berührung mit den Fingern oder mit Gegenständen gefährdet werden kann.

Einstellen der Slave-Adressen

Das Einstellen der Slave-Adressen geschieht über Jumper, die als Lötbrücken ausgeführt sind. Möchte man die Adresse verändern, ist zunächst die Verbindung der ab Werk eingestellten Jumper zu entfernen. Hierzu

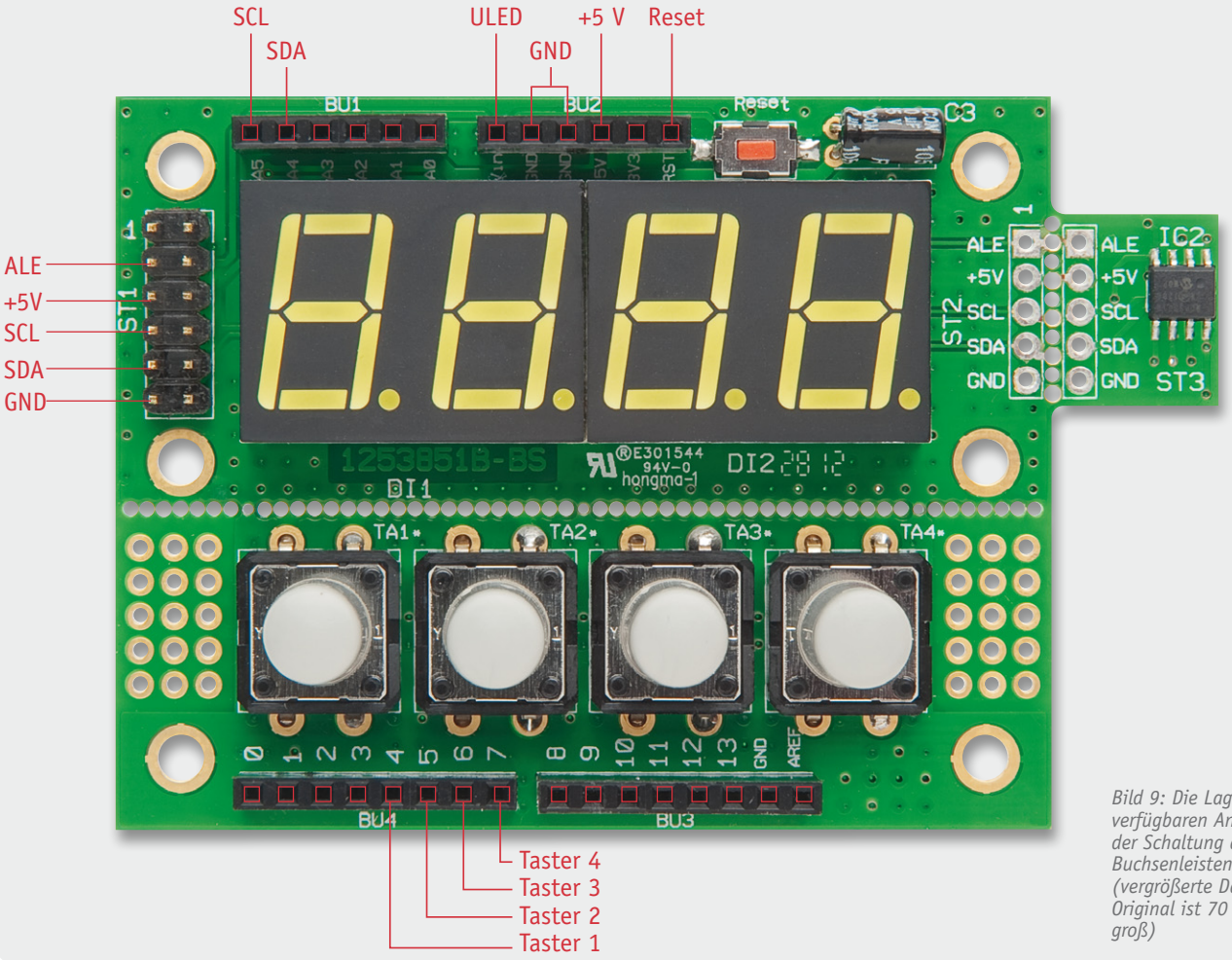


Bild 9: Die Lage der verfügbaren Anschlüsse der Schaltung an den Buchsenleisten (vergrößerte Darstellung, Original ist 70 x 55 mm groß)

wird die Verbindung mit einem scharfen Messer o. Ä. vorsichtig aufgetrennt. Hierbei ist darauf zu achten, dass keine benachbarten Leiterbahnen beschädigt werden. Die Verbindung an einem Jumper wird durch Herstellen einer Lötzinnbrücke realisiert.

Separate Spannungsversorgung der 7-Segment-Anzeigen

Die gesamte Elektronik wird im Normalfall über den Anschluss +5 V mit Spannung versorgt. Falls die Spannungsquelle nicht genügend Strom (max. 170 mA) liefern kann, muss die Versorgung der 7-Segment-Anzeigen über einen separaten Anschluss (ULED) erfolgen. Die Spannung an diesem Eingang muss höher als die 5-V-Versorgungsspannung sein. Eine alleinige Spannungsversorgung der gesamten Elektronik über den Anschluss ULED ist nicht möglich.

Arduino-Shield

Bei dem Betrieb mit dem Arduino-Board sind, wie beim Nachbau beschrieben, alle Buchsen/Stiftleisten zu bestücken. Wichtig hierbei ist, dass die Platine richtig herum auf das Arduino-Board aufgesteckt wird. Als Orientierung kann dabei die deutlich hervorstehende USB-Buchse des Arduino dienen. Bild 10 zeigt die exakt zusammengesteckte Kombination. Es ist besonders darauf zu achten, dass alle Stiftleisten in die entsprechenden Buchsenleistenkontakte des Arduino-Boards „greifen".

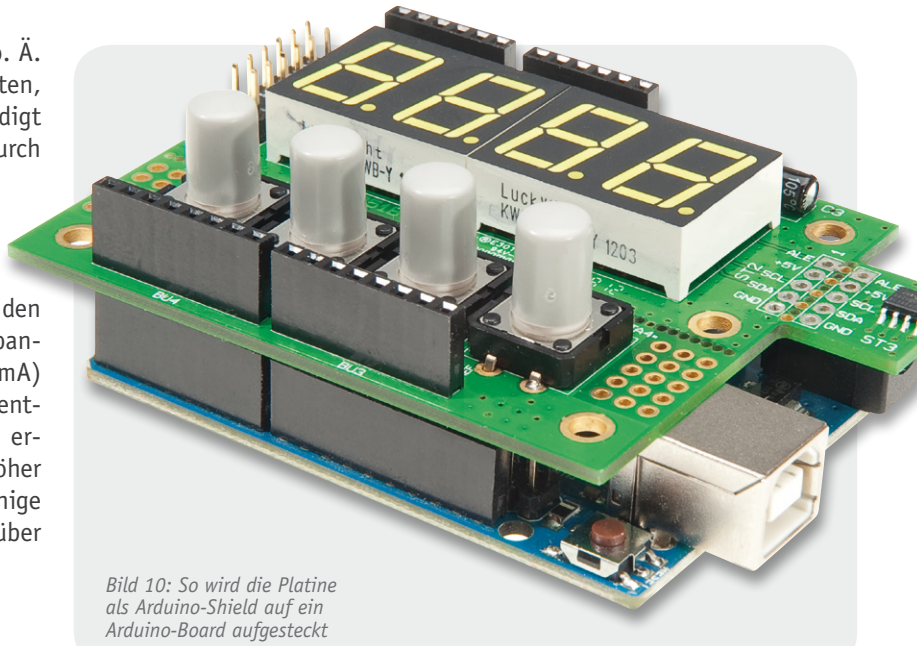


Bild 10: So wird die Platine als Arduino-Shield auf ein Arduino-Board aufgesteckt

USB-I²C-Adapter

Eine weitere mögliche Anwendung ist der Betrieb mit dem ELV-USB-I²C-Adapter (Bild 11). Hierbei erfolgt die Steuerung von einem PC aus. Wie dies im Einzelnen erfolgt, ist in der Anleitung des I²C-Adapters beschrieben. Zum Testen der Schaltung kann das kostenlose USB-Testtool (USB-I²C-Demosoftware) [4] verwendet werden. Die Verbindung erfolgt, wie in Bild 11 zu sehen, über den Anschluss ST1 auf der Platine mit folgender Zuordnung:

I²C-USB-Adapter	<->	I²C-7-Segment-Anzeige
+5 V	<->	+5 V
SCL	<->	SCL
SDA	<->	SDA
GND	<->	GND

Die abgebildete Temperatursensorplatine ist natürlich auch hier nicht zwingend erforderlich. Zudem kann diese Platine auch ohne die Anzeigenplatine direkt mit dem I²C-Bus-Adapter verbunden werden. In diesem Anwendungsbeispiel kann der Alarmausgang (ALE) nicht genutzt werden, da hierfür die notwendige Rückleitung beim USB-I²C-Adapter nicht vorhanden ist. Da der USB-I²C-Adapter bis zu 500 mA liefern kann, ist eine zusätzliche Spannungsversorgung nicht erforderlich.

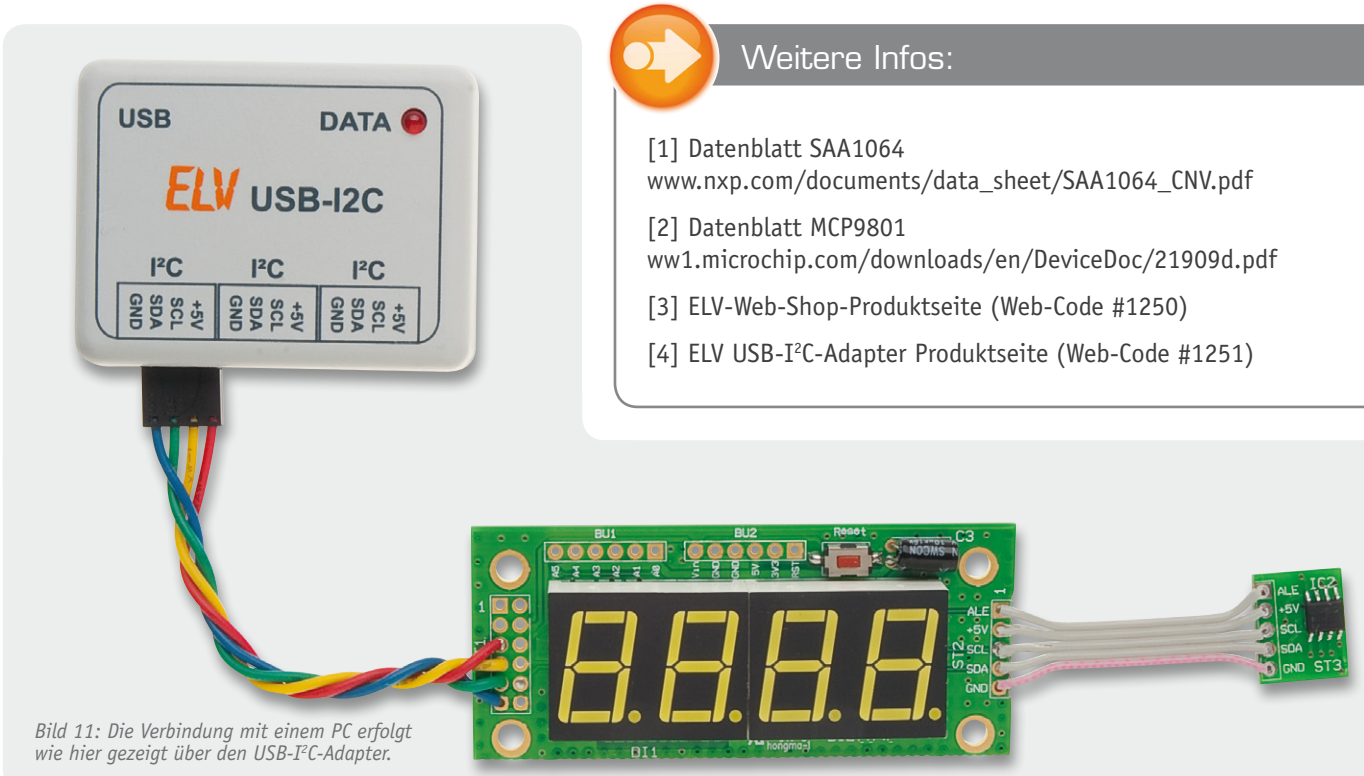


Bild 11: Die Verbindung mit einem PC erfolgt wie hier gezeigt über den USB-I²C-Adapter.

Weitere Infos:

[1] Datenblatt SAA1064
www.nxp.com/documents/data_sheet/SAA1064_CNV.pdf

[2] Datenblatt MCP9801
ww1.microchip.com/downloads/en/DeviceDoc/21909d.pdf

[3] ELV-Web-Shop-Produktseite (Web-Code #1250)

[4] ELV USB-I²C-Adapter Produktseite (Web-Code #1251)