



# Arduino steuert einen intelligenten Adventskalender

Auch in diesem Jahr wird in vielen Familien in der Adventszeit wieder ein besonderer Kalender die Wohnzimmerwand zieren. Die in diesem Beitrag von unserem Leser Engelbert Wedekind vorgestellte Variante eines Adventskalenders soll Anregung sein für eine Erweiterung seiner ursprünglichen Funktion. Die wichtigste Aufgabe des Kalenders, jeden Tag hinter einem Türchen eine kleine Überraschung zu finden, soll nun auch noch akustisch unterstützt werden. Der Adventskalender soll „musikalisch“ werden.



**Engelbert Wedekind**

hat für seinen Beitrag zum Leserwettbewerb einen 200-Euro-Warengutschein erhalten!

## Wie es begann

Einen Adventskalender wird sicher jeder nach seinen individuellen Bedürfnissen gestalten, wenn er etwas Besonderes sein soll. Für ambitionierte Elektroniker soll hier die Idee für einen „artgerechten“ Adventskalender zum Nachbau vorgestellt werden. Er hat nicht nur die 24 Türchen mit den dahinter versteckten Süßigkeiten zum Inhalt. In diesem Fall soll er durch eine intelligente Steuerung ergänzt werden.

Allein die Aufzählungsversuche solcher Lösungen würden den Rahmen dieses Artikels sprengen. Hier soll eine realisierte Variante vorgestellt werden, die genügend Spielraum für eigene Vorstellungen bietet.

Hauptanliegen ist es, das Öffnen der (richtigen!) Tür zu überwachen und dann ein Adventslied mit Hilfe des integrierten MP3-Players abzuspielen. Das „richtig“ geöffnete Fach wird noch zusätzlich beleuchtet. Zur Abschreckung allzu Neugieriger löst ein „versehentlich“ falsch geöffnetes Fach einen kurzen Warnton aus.

Um die akustische Wiedergabe eines Adventsliedes zum richtigen Zeitpunkt zu ermöglichen, soll jede Tür



des Kalenders in ihrer „Nicht-zu-Stellung“ überwacht werden. Wird z. B. am 1. Dezember das Türchen mit der Türnummer 1 geöffnet, wird dies über einen Reflexkoppler CNY70 erkannt und die Freigabe erteilt, automatisch ein Adventslied abzuspielen. Mit dem erzeugten Startbefehl nach Öffnen der Tür wird zunächst die abgespeicherte Titelnummer an einen Arduino per UART gesendet. Im Programm des Arduino wird die zugehörige Datei, die auf einer SD-Card im MP3-Player abgespeichert ist, geöffnet, die Daten der MP3-Datei eingelesen und an die Audio-Baugruppe ASA1 [1] gesendet. Diese besteht aus einem MP3-Player und dem Stereoverstärker. Es sind nur noch die beiden Lautsprecher für den linken und rechten Kanal anzuschließen. Die Baugruppe ASA1 ist zusammen mit der RTC-Uhr-Baugruppe auf einen Arduino Uno aufgesteckt (Bild 1).

In den vergangenen Jahren wurden bereits zwei Vorgängerversionen des Kalenders konstruiert und aufgebaut.

Variante 1 ist wie ein Wandkalender aufgebaut. Der Korpus des Kalenders wurde auch hier schon aus naturbelassenen Sperrholzteilen gefertigt. Er verfügt über ein Display, welches neben aktuellem Datum (interne RTC-Uhr, DS1307) auch Bedienungshinweise anzeigt. Der Name des Titels wird angezeigt, wenn die richtige Tür am richtigen Tag geöffnet wird. Bis auf den MP3-Player (hier noch ein SOMO 14D [2]) wurden die Steuerungsteile inklusive Netzteil (5 V und 3,3 Vdc) auf Laborkarten aufgebaut. Das geöffnete Fach der „richtig“ geöffneten Tür (Nummer auf der Tür stimmt mit aktuellem Datum überein) wird intern mit einer ultrahellen LED beleuchtet, was bei Kerzenschein im abendlichen Halbdunkel des Dezembers einen sehr effektvollen, scheinwerferähnlichen Lichtkegel im offenen Fach erzeugt.

Variante 2 ist ein ebenfalls aus verleimten Sperrholzteilen aufgebautes Modellhaus mit einem Spitzdach und äußerlich farbiger Gestaltung. Dieses sehr viel aufwändigere Modell hat keine Türen, sondern Einschubfächer, die in das Modellhaus eingeschoben werden. Die Größe der Kästen ermöglicht es, außer Süßigkeiten auch kleineres Spielzeug mit zu verschenken. Da kleinere Kinder noch keine Zahlen lesen können, aber selbstständig ihren Kalender öffnen können sollten, wurden die LEDs von Variante 1 dazu benutzt, den aktuell zu ziehenden Einschub zu „markieren“. Als besonderes Highlight wurde hier noch ein Schornstein integriert, in den eine Schornsteinheizung aus der Modellbahntechnik eingebaut wurde. Wird nun das „richtige“ Kästchen (s. o.) gezogen, wird während des Abspielens des Adventsliedes noch ein wenig Qualm (spezielles Verdampferöl) aus dem Schornstein erzeugt.

Als MP3-Player wurde hier das MP3-Soundmodul MSM2 von ELV mit nachgeschaltetem Verstärker eingesetzt. Nachteil dieser Baugruppe für den Einsatz im Kalender ist es, dass nur 20 Titel durch Schließen eines Schaltkontaktes ausgewählt werden können. Damit müssen immer 4 Lieder doppelt zugeordnet werden. Um diesen Nachteil auszugleichen, wurde die Baugruppe nach dem ersten Review gegen eine neue, per UART ansteuerbare MP3-Player-Baugruppe ausgetauscht. Die neue Infrarot-Fernbedienungs-Baugruppe IRE8 wurde bei diesem Update der Steuerung ebenfalls nachgerüstet. Nähere Informationen zu den genannten Baugruppen findet man unter [3] und [4].

Die dritte Variante des Kalenders wurde nun so konzipiert, die bisher gesammelten Erfahrungen zu nutzen und zur einfacheren Reproduzierbarkeit geeignete und am Markt verfügbare TWI-Baugruppen einzusetzen. Aufgrund der aktuellen und nahezu weltweiten Verbreitung fiel die Wahl auf Arduino-kompatible und per I<sup>2</sup>C-Bus koppelbare Baugruppen. Der I<sup>2</sup>C-Bus wird aus lizenzrechtlichen Gründen auch als TWI-Bus bezeichnet und diese Bezeichnung in den weiteren Erläuterungen mit verwendet. Aktuelle Informationen zum Arduino Uno und der großen Anzahl von Baugruppen der Arduino-Welt finden sich z. B. unter [5].

## Schaltungsbeschreibung

Die Steuerung des Adventskalenders (Bild 2 zeigt den Übersichts-Schaltplan) ist in zwei Hauptteile untergliedert. Aufgrund des Aufbaus

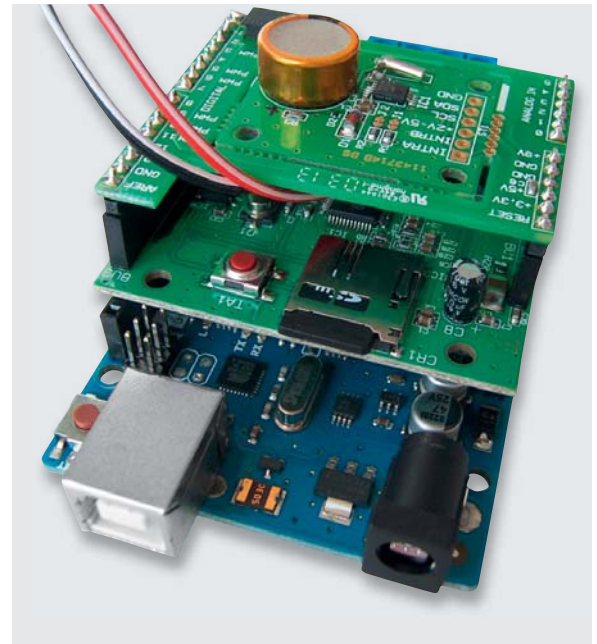


Bild 1: Der Arduino Uno mit den ELV-Shields ASA1 und I<sup>2</sup>C-RTC

aus überwiegend konfektionierten Modulen soll hier im Wesentlichen nur deren Zusammenwirken für die Erfüllung der wichtigsten Aufgaben erklärt werden. Den ersten Teil bilden ein Arduino Uno mit aufgesteckter Audio-Baugruppe ASA1 und einer RTC-Uhr-Baugruppe. Die Verbindung zum zweiten Teil der Steuerung wird nur über die UART-Schnittstelle hergestellt.

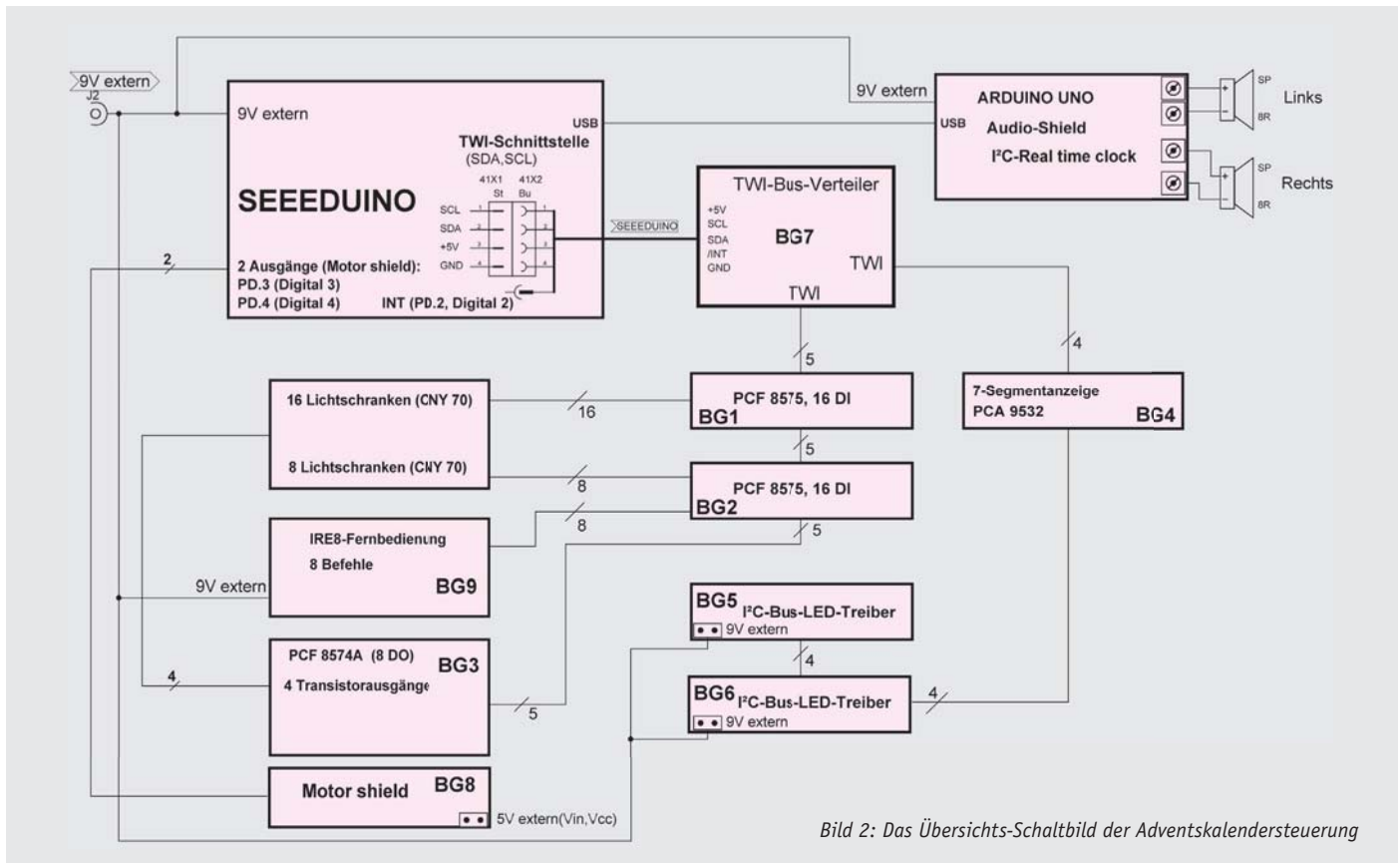
Nähere technische Informationen zu diesen Baugruppen sind unter [1], [5] und [6] zu finden.

Als Steuercontroller für die Baugruppen des zweiten Teils der Steuerung wurde ein Seeeduno der Firma Seeed Studio [7] (Bild 3) ausgewählt. Er meldet sich in der Arduino-IDE als Arduino Uno. In der aktuellen Version 3.0 sind die 2 erforderlichen Schnittstellen (I<sup>2</sup>C und UART) bereits auf separate Steckkontakte herausgeführt. Diese Schnittstellen werden jeweils über ein passend erhältliches Adapterkabel angeschlossen. Links zu Lieferanten für die Baugruppe und die Adapterkabel sind unter [8] und [9] zu finden. Die beiden Schnittstellen werden für die Kommunikation zwischen den peripheren Baugruppen und dem Seeeduno einerseits sowie dem Arduino Uno und dem Seeeduno andererseits benutzt.

Mit Hilfe von zwei Eingangsbaugruppen werden die Schaltzustände der 24 Reflexkoppler CNY70 erfasst

### Anmerkung der Redaktion:

Der Beitrag von Herrn Wederkind ist sehr umfangreich, er enthält allein 15 Schaltbilder, er würde deshalb den maximalen Rahmen eines ELVjournal-Artikels sprengen. Deshalb haben wir uns entschlossen, eine gekürzte Fassung des Artikels hier abzdrukken und den Artikel in der vollen Länge als Download auf der Webseite des ELVjournals unter Webcode #1334 kostenlos zur Verfügung zu stellen.



und dadurch die Stellung der Türen überwacht. Die Ausgabebaugruppe aus Bild 4 steuert das Auslesen der CNY70, indem die LEDs der Koppler nur zum Zeitpunkt des Auslesens mit Spannung versorgt werden. Damit kann die Gesamtstromaufnahme der Steuerung erheblich gesenkt werden. Sowohl die Größe der Kollektorwiderstände der Fototransistoren der CNY70 als auch die Vorwiderstände für die Einstellung des Stroms durch die Infrarotdiode für einen geeigneten Arbeitspunkt der Koppler wurden anhand der konkreten Reflexionsbedingungen an den Kanten der überwachten Türen am fertigen mechanischen Aufbau experimentell ermittelt.

Sowohl die Eingangsbaugruppen aus Bild 5 und 6 (das Datenblatt für den PCF 8575 findet sich unter [10]) als auch die Ausgangsbaugruppe aus Bild 4 werden über den Anschluss am TWI-Verteiler (Bild 7) mit Spannung versorgt. Über die Eingangsbaugruppe in Bild 6 werden über

die 8 nicht für die Koppler verwendeten Eingänge die 8 Steuerbefehle der Infrarot-Fernbedienung eingelesen. Näheres zur Auswahl der Fernbedienung und zur Einstellung der Parameter der Baugruppe ist in [3] nachzulesen.

Mit einem akustischen Signalgeber SP1, an die Ausgangsbaugruppe in Bild 4 angeschlossen, wird noch eine Warnung ausgegeben, wenn die „falsche Tür“ – sprich Türnummer und aktueller Tag stimmen nicht überein – geöffnet wird.

Bild 8 zeigt, wie zwei Reflexkoppler CNY70 über der Türkante in einem Doppelfach eingebaut sind. Bild 9 zeigt die fertig verdrahtete Fachreihe 1, bestehend aus 3 Doppelfächern. Die hier noch nicht beschalteten Ein- und Ausgänge der montierten Eingangs- und LED-Baugruppe werden dann von den darüber liegenden Fachreihen nach unten verdrahtet.

Die I²C-LED-Ausgangsbaugruppen, wie in Bild 10 zu sehen, übernehmen die Ansteuerung der Beleuchtung des geöffneten Faches durch superhelle LEDs [11]. Diese LEDs werden mittig im Fachteil von oben durch eine 3-mm-Bohrung hindurchgesteckt. Die Beleuchtung des Faches wird aber nur aktiviert, wenn die richtige Tür am richtigen Tag geöffnet wird. Es ist also immer nur eine LED eingeschaltet. Diese LED-Baugruppen werden nur signalmäßig an den TWI-Bus angeschlossen. Die Versorgung mit 9 Vdc erfolgt über die Klemmen auf der Baugruppe direkt vom Verteiler des speisenden 9 Vdc-Netzgerätes. Der maximal mögliche Ausgangsstrom für die LEDs wird auf der Baugruppe durch einen Widerstand vorgegeben.

Über die Baugruppe aus Bild 11 werden neben dem aktuellen Tag noch Quittierungen der Steuerung nach Befehlseingaben angezeigt. Informationen zum eingesetzten PCA 9532 finden sich unter [12].

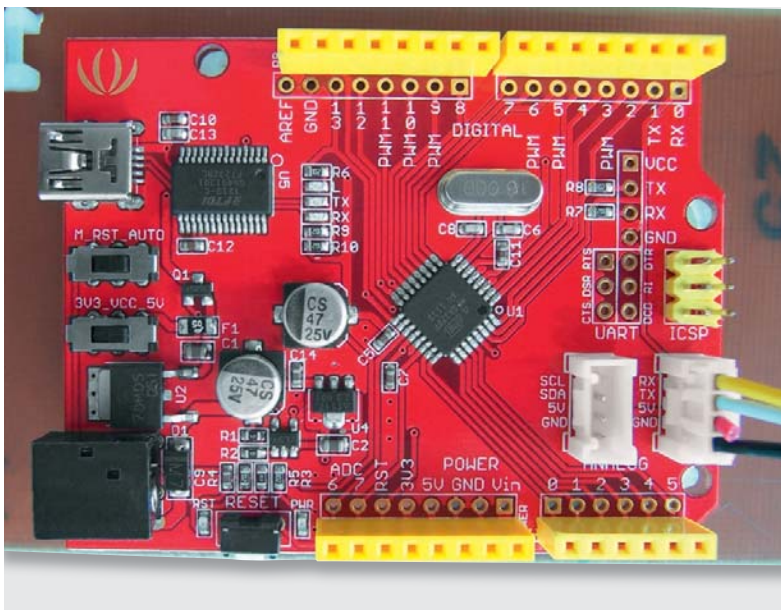


Bild 3: Der Seeeduino 3.0 wird als Steuerung eingesetzt.

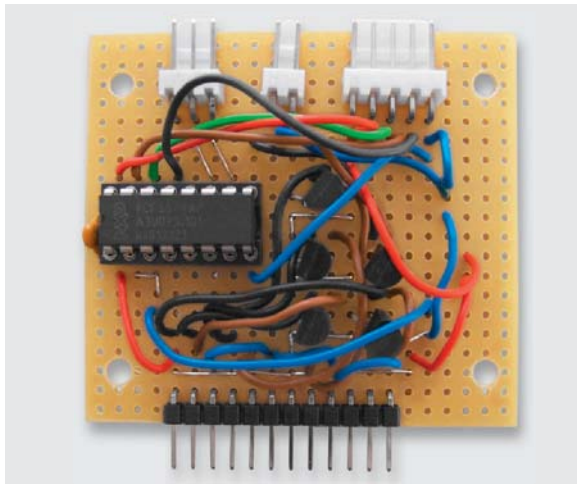


Bild 4: Die Ausgangsbaugruppe versorgt auch die Infrarot-Sende-dioden der CNY70-Optokoppler.

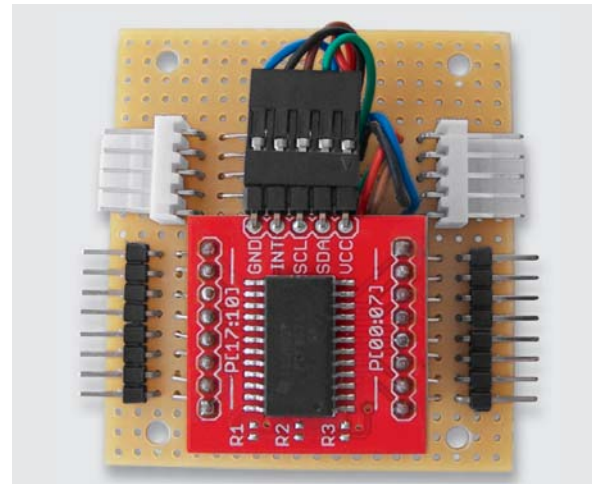


Bild 5: Die Eingangsbaugruppe 1 überwacht den Zustand der Opto-koppler und der Türen.

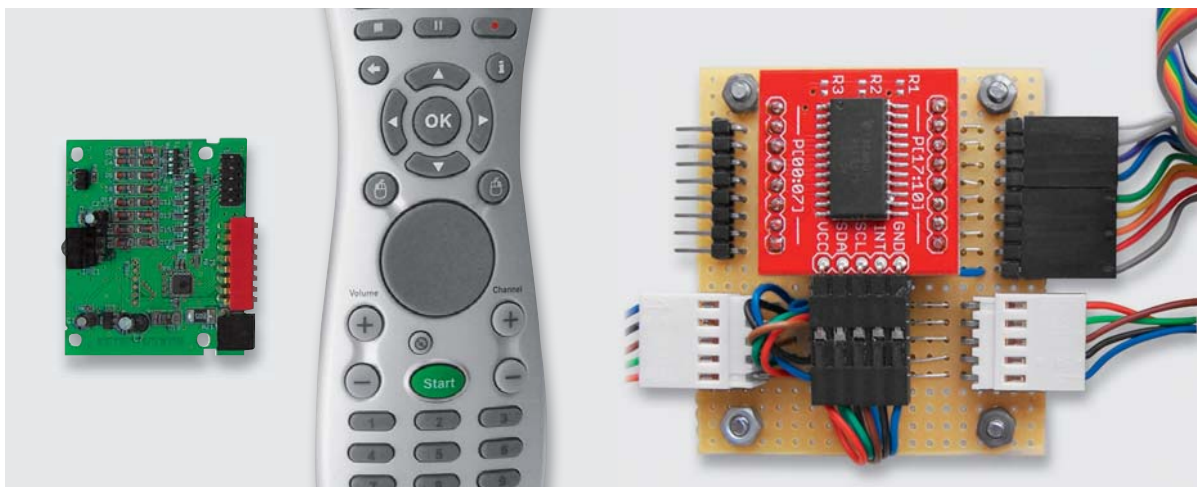


Bild 6: Die Infrarot-Fernbedienungs-Baugruppe IRE 8 mit Anschluss an Eingangsbaugruppe 2

Es ist bei allen Adresseinstellungen der TWI-Bau-gruppen darauf zu achten, dass es nicht zu Über-schneidungen bei der TWI-Adressierung kommt, da mitunter gleiche Adressbereiche von mehre-ren TWI-ICs benutzt werden. Die im Stromlauf-plan angegebenen TWI-Adressen werden an den Eingangsbaugruppen aus [Bild 5 und 6](#) sowie der Transistor-Ausgangsbaugruppe aus [Bild 3](#) und der Anzeigebaugruppe aus [Bild 11](#) fest verdrahtet. Die Adresseinstellung an den I<sup>2</sup>C-LED-Baugruppen ([siehe Bild 9](#)) erfolgt mit Jumpern.

Für die Inbetriebnahme oder erste Tests der TWI-Baugruppen des doch umfangreichen Aufbaus noch ohne Controller ist die USB-I<sup>2</sup>C-Interface-Baugruppe [\[13\]](#) zu empfehlen.

Auf der Oberseite des Kalenders ist ein Drehteller mit aufgesteckten Holzfiguren montiert ([Bild 12 und 14](#)). Dieser Drehteller wird mit einem Getriebemo-tor angetrieben, der während der Wiedergabe eines Adventsliedes mit einer vorgegebenen PWM über ein spezielles Motor-Shield ([Bild 13](#)) eingeschaltet wird. Die Geschwindigkeit kann über das PWM-Signal noch „feinjustiert“ werden. Nähere Informationen zum Motor-Shield findet man unter [\[14\]](#).

In [Bild 14](#) ist die Rückseite des Kalenders mit den montierten Baugruppen zu sehen. Hier sind der TWI-

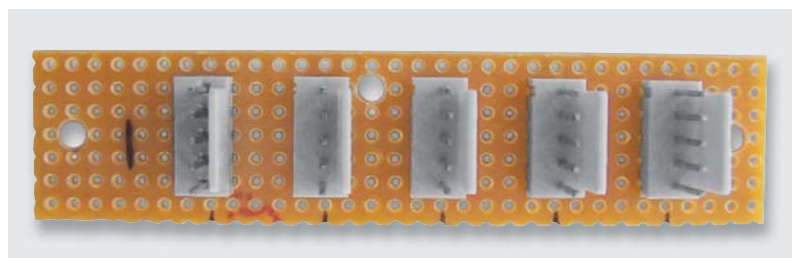


Bild 7: Der TWI-Bus-verteiler sorgt für die Signal- und Spannungsverteilung.



Bild 8: Ansicht der Anordnung von 2 Reflexkopplern an den Türkanten eines Doppelfaches

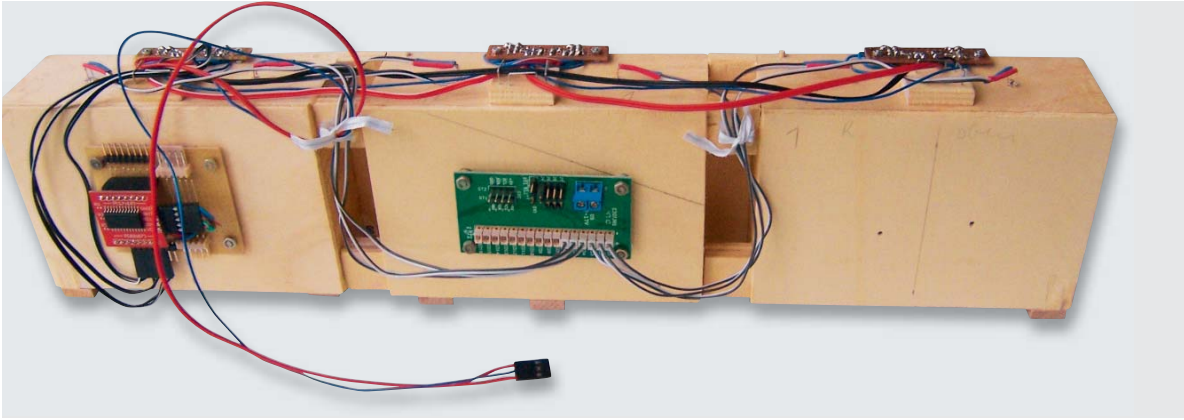


Bild 9: Ansicht einer fertig verdrahteten Fachreihe

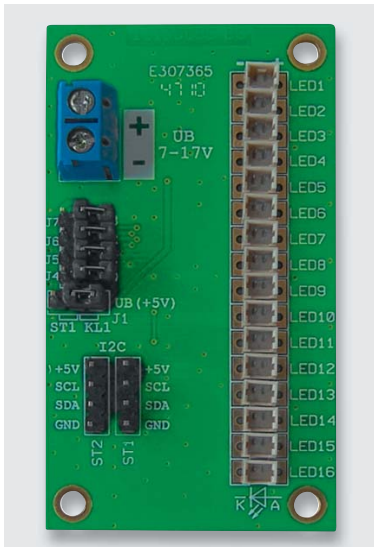


Bild 10: Die I²C-LED-Ausgangsbaugruppe

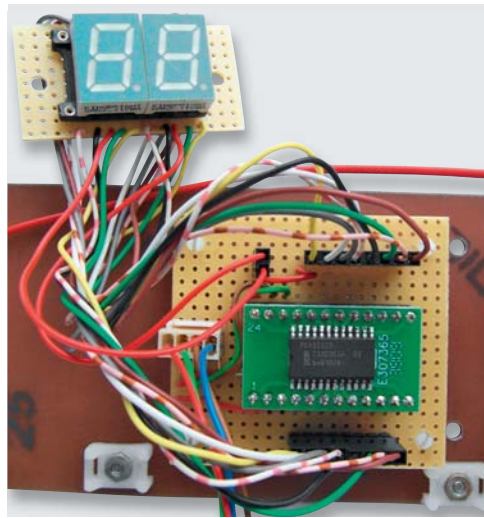


Bild 11: Ausgangsbaugruppe 3, Anzeige des Datums und weiterer Meldungen



Bild 12: Getriebemotor mit Drehscheibe und aufgesteckten Pyramidenteilen

Bus und die Spannungsversorgung schon verdrahtet sowie die Steuerplatine mit dem Arduino-Aufbau, dem Seeduino und den restlichen Baugruppen zu sehen.

Eine genauere Erläuterung des Zusammenwirkens der Baugruppen erfolgt in der steuerungstechnischen Beschreibung des Projekts [15], die komplett über den Artikel-Download im Web-Bereich verfügbar ist. Hier finden sich dann auch Codebeispiele zur Programmierung des Systems.

## Erweiterungen

Zum Ende noch ein Ausblick auf mögliche Erweiterungen des vorgestellten musikalischen Adventskalenders. Die Ausgabe von Meldungen und Störungen

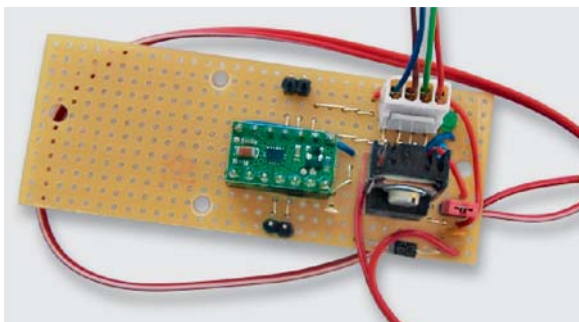


Bild 13: 5-V-Netzteil und Motorantriebsbaugruppe DRV 8835

im Ablauf kann über ein zusätzliches Display erfolgen. Dazu zählen z. B. Informationen wie „keine SD-Card gesteckt“, „Arduino nicht betriebsbereit“, „keine \*.mp3-Dateien auf der SD-Card“, „Zum Start nicht alle Türen geschlossen“ oder auch die Ausgabe der Dateinamen aller abgespeicherten MP3-Files. Hier könnten dann auch die mitgezählten „heimlichen und unerlaubten Öffnungsversuche von Neugierigen“ gemeldet werden. Wem ein Zeilen-Display nicht mehr zeitgemäß genug ist, kann der Steuerung auch noch Erweiterungsbaugruppen in Form von Grafik-Displays oder zusätzlichen Schnittstellen aus dem Fundus der Arduino-Welt für weitere Funktionen hinzufügen. Damit können solche Informationen direkt per WLAN oder Bluetooth zum Smartphone oder zum Hausautomationssystem gemeldet werden. Hier kann sich jeder ambitionierte Elektroniker nach seinen Wünschen und Möglichkeiten einbringen.

Da viele Aufgaben der Steuerung mit inzwischen vielfältig verfügbaren konfektionierten Baugruppen zu lösen sind, sollte der Nachbau der Steuerung keine größeren Probleme bereiten. Durch den Einsatz weiterer „konfektionierter“ Lösungen können auch die restlichen, hier noch mit diskreter Technik aufgebauten Ansteuerungen nachbausicher vereinfacht werden.

Die eigentliche Herausforderung und damit der Spaß am Nachbau des prinzipiellen Gedankens des musikalischen Adventskalenders stellt eindeutig der mechanische Aufbau des Adventskalenders dar. Die hier vorgestellte Variante soll dazu nur eine erste Anregung bieten. So ein musikalischer Kalender wird mit Sicherheit kein „Sonntagnachmittagsprojekt“, aber der Aufwand für eine solch außergewöhnliche Geschenkidee lohnt sich.

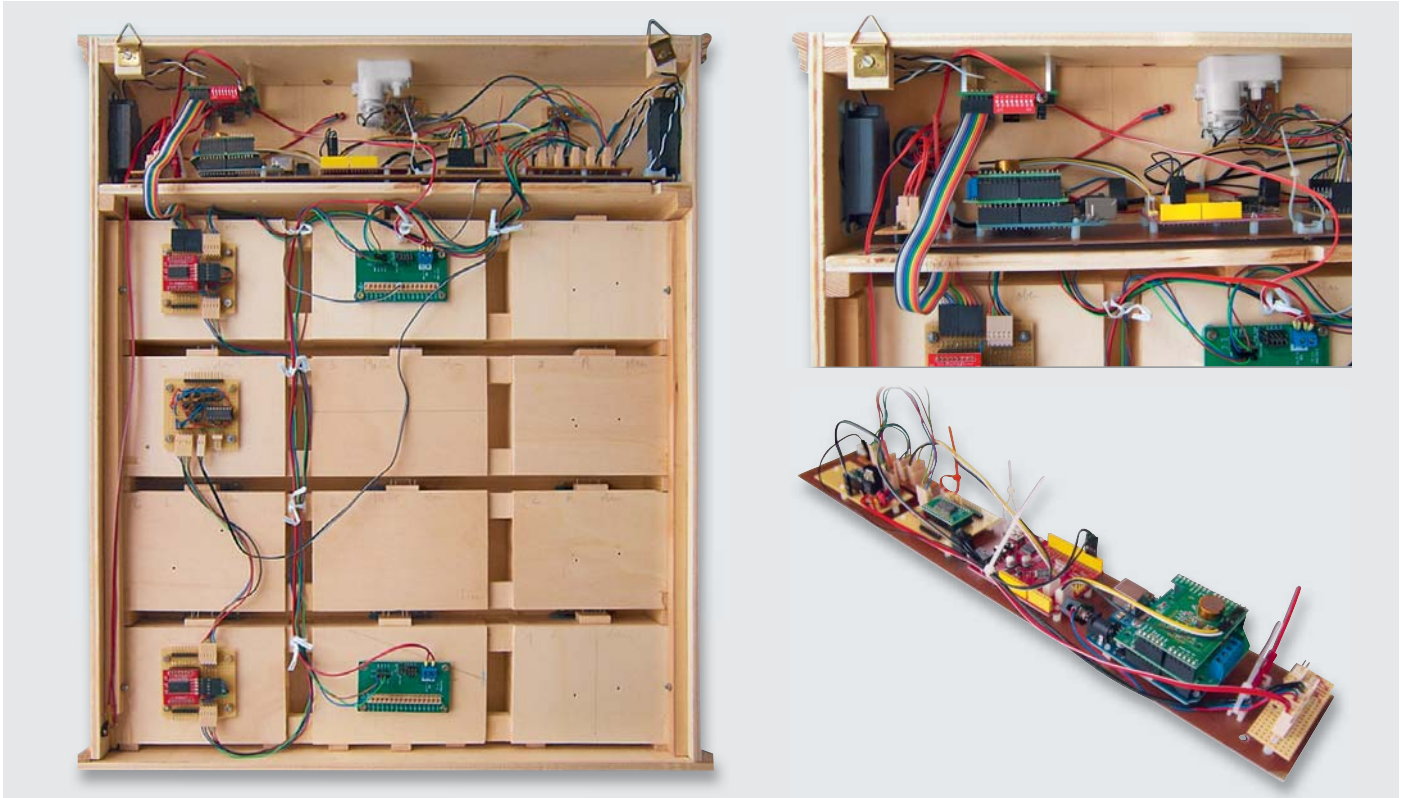


Bild 14: Die Rückseite des Kalenders, rechts die Montage der beiden Controllerboards auf der Steuerplatine und die Ansicht der gesamten Steuerplatine

# Beschreibung der Steuerung des Adventskalenders

## Programmaufbau und RTC

Zur Wiedergabe der auf einer SD-Card abgespeicherten Adventslieder wurde das auf [1] zur Verfügung stehende Projekt für den ASA1 in die Arduino-IDE (Version 1.0.5.-r2) komplett eingebunden. Die Initialisierung der Baugruppen und die Hauptschleife wurden in dem schon vorgegebenen Programmaufbau entsprechend ergänzt und noch fehlende Teile neu erstellt [16].

Die RTC-Uhr, aufgesteckt auf den ASA1, liefert das aktuelle Datum zum Vergleich mit der Nummer der jeweils geöffneten Tür. Die RTC-Uhr muss bei der ersten Inbetriebnahme einmalig mit aktuellen Werten „gestellt“ werden. Dazu ist im Initialisierungsteil des Arduino-Projektes (hier in der Version ASA1\_V3 [16]) entsprechender Code vorbereitet. Voraussetzung ist die korrekte Einbindung der von [1] ladbaren Bibliothek und die Übernahme des Beispielsketches dieser RTC-Baugruppe in das Projekt vom ASA1. Die Einstellparameter für die Funktion DateTime() sind entsprechend dem konkreten Zeitpunkt des Uploads des Projektes in den Arduino Uno vorzunehmen. Für den nächsten Upload muss dann diese Einstellung aktualisiert werden, oder man kommentiert diese Codezeilen aus. Die Stützung der RTC-Uhr durch den Goldcap reicht nach bisherigen Tests durchaus für mehrere Tage. Da der Kalender täglich benutzt wird,

sollte das korrekte Datum nach einmaliger Einstellung vor dem Start der Adventszeit sicher verfügbar sein, auch wenn die Steuerung zwischenzeitlich ausgeschaltet wird. Wird der Kalender dann im darauf folgenden Jahr wieder benutzt, bitte die RTC-Uhr neu stellen.

## Seeeduo-Software

Die aktuelle Software für den Seeeduo wurde, aus der Historie heraus, mit Hilfe eines kommerziellen Compilers von HP InfoTech (CVAVR CodeVision, aktuelle Version V3.14, Link zu einer Demoversion unter [17]) erstellt und in Atmel Studio (aktuelle Version 6.2) mit einem AVR-Dragon-Emulator getestet. Informationen zu Atmel Studio und zum Emulator AVR Dragon findet man auf der Homepage der Firma Atmel [18]. In Atmel Studio steht auch ein eigener, kostenfreier GCC-Compiler nach GNU-Lizenzierung zur Verfügung.

Bei Verwendung der Arduino-IDE auch für den Seeeduo ist die Programmkontrolle mit gewissen Einschränkungen ebenso möglich. Über die USB-Schnittstelle können während des Tests des Programms wie üblich Debugging-Informationen ausgegeben werden, allerdings ohne direkte Kontrolle der gesamten Hardware des Controllers und aller Variablen, die nicht gerade per Print-Anweisung über die serielle Schnittstelle ausgegeben werden. Die UART-Schnittstelle wird zusätzlich schon für die Kommunikation der beiden Controller verwendet. Ausgaben, die nicht der Kommunikation beider Controller dienen, müssen deshalb im Programm überwacht werden. Da auch nicht druckbare Zeichen über die UART-Schnittstellen ausgetauscht werden sollen, empfehle ich, zur Ausgabe der Debug-Informationen Programme wie HTerm zu benutzen [19].



Die einzelnen Teilaufgaben der Software im Seeeuino sollen mit Hilfe von Funktionen gelöst werden, die in einer Hauptschritt看ette in einem logisch begründeten Ablauf aufgerufen werden. Im Initialisierungsteil des Programms sind zunächst alle notwendigen Einstellungen für die Ports und die Schnittstellen vorzunehmen. Entsprechend Stromlaufplan sind die Ports zu deklarieren und die UART-Schnittstelle mit nachfolgenden Parametern zu initialisieren:

Baudrate: 38.000  
Daten-Bits: 8 Data-Bit  
Stopp-Bits: 1 Stopp-Bit  
Parität: keine Parität

Die UART-Schnittstelle des Arduino Uno ist mit den gleichen Parametern eingestellt (s. a. ArduinoSketch ASA\_V3.ino [16]). Die etwas höhere Taktrate sorgt u. a. mit dafür, dass während des Abspielens des Titels vom Seeeuino an den Arduino Uno gesendete Lautstärkeänderungen keine hörbaren Unterbrechungen des Abspielvorganges verursachen. Weiterhin ist eine TWI-Schnittstelle im Mastermodus zu initialisieren. Als Bitrate ist 100 kHz, entsprechend den Vorgaben durch die eingesetzten TWI-ICs, vorgesehen. Um das Motor-Shield für den Getriebemotor der Drehscheibe mit dem kleinen Pyramidenaufbau ansteuern zu können, wurde der Timer 2 im Non-inverted PWM-Modus eingestellt (PWM-Ausgang OC2B, PORTD.3 als Output). Die PWM-Frequenz soll laut Datenblatt des Motor-Shields auf < 250 kHz eingestellt werden. Über das vom Seeeuino ausgegebene PWM-Impulsverhältnis an den IN2 des Motor-Shields kann die resultierende Geschwindigkeit des Getriebemotors bei Bedarf weiter verringert werden. Zusätzlich zum Impulsverhältnis wird auch die Drehrichtung vorgegeben. Die Einstellung der Geschwindigkeit selbst hängt aber im Wesentlichen vom eingesetzten Getriebemotor und den persönlichen Vorstellungen ab. Um den Getriebemotor über den Steuereingang IN1 mit dem am PWM-Eingang IN2 anliegenden Impulsverhältnis ein- oder auszuschalten, ist PORTD.4 als Output zu initialisieren. Als Letztes sind die TWI-Bausteine für den Startzustand zu initialisieren. Die Eingangs-, Ausgangs- und LED-Baugruppen sind entsprechend den Vorgaben im jeweiligen Datenblatt und den sonstigen Wünschen des Anwenders zu programmieren. Die TWI-ICs für die Anzeige des Tages und die Beleuchtung der Fächer sollten dabei zur Anpassung an die Umgebungshelligkeit im PWM-Modus initialisiert werden. Wird zur Programmierung des Seeeuino die Arduino-IDE eingesetzt, sollten sich dafür Bibliotheken und ihre Beispielsketches aus dem Internet nachnutzen lassen.

### Initialisierung und Deklaration von Variablen

Im Hauptteil des Programms sind eine Reihe von Variablen und Strukturen global zu deklarieren und zu initialisieren. An dieser Stelle sollen nur die für die Umsetzung der Idee des Kalenders wichtigsten Variablen und Strukturen erfasst werden. Weitere Hilfsvariablen sind wie immer entsprechend zu ergänzen.

Um die Suchalgorithmen im Programm zu vereinfachen, soll als wichtigstes Prinzip die Verknüpfung

der Türnummern mit den Masken der zum Fach zugehörigen LEDs und der Titelnummer des Adventsliedes auf der SD-Card in der Audio-Baugruppe ASA1 eingeführt werden. Als Erstes wird der Anwender die Anordnung der Nummern 1 bis 24 auf den 24 Türen des Adventskalenders nach seinen Wünschen vornehmen. Damit steht fest, welche Türnummer zu welcher (physischen) Tür und damit zu welchem Bit an den Eingangsbaugruppen, eingelesen in Struktur uc8\_tueren[4], gehört. Die verteilten Türnummern sind jetzt in der Struktur sc\_TuerNummer[4] an der entsprechenden Stelle in den 3 Byte für Tür 1 bis 24 einzutragen. An der gleichen Stelle ist in Struktur struct FileNumber[4] die Nummer des für diesen Tag gewünschten Titels einzutragen. Mit dieser Festlegung kann man mit dem gefundenen Tag (physischer Eingang ist EIN) aus der Struktur sc\_TuerLed[4]. Leds[8] die Einschaltmaske (int-Werte) für die über das Fach zugeordnete LED und aus der Struktur struct FileNumber[4] die Titelnummer in der gleichen Schleife mit dem gleichen Index auslesen. Das spart Zeit und Speicherplatz.

Es steht dem Anwender natürlich frei, weitere Sound-Schnipsel oder andere MP3-Dateien zur Ausgestaltung des Ablaufs am Kalender zur Verfügung zu halten und deren Abspielen im Steuerungsablauf an der gewünschten Stelle einzubinden. Im Anhang „Softwarebeispiele“ sind unter Punkt 1 beispielhaft Deklarationen und Initialisierungen zu den oben genannten Strukturen angegeben.

In einem weiteren Schritt ist das Maskenarray für die Anzeige des Tages über den PCA 9532 zu initialisieren (siehe Bild 11). Die 7 Segmente der Anzeige sind den Ausgangsbits des PCA 9532 entsprechend dem Stromlaufplan zugeordnet. Für die Speicherung der Ansteuermasken ist ein Array mit int-Werten zu deklarieren (s. Anhang „Softwarebeispiele“) und mit den Bitmasken zu initialisieren. Die angegebenen Masken stehen dabei für die Darstellung der Ziffernfolge 0...9. Die Maske für jede Ziffer wird auf Basis der Angaben im Datenblatt des Herstellers erstellt [20]. Soll das Segment eingeschaltet werden, ist die Bitkombination „10“ in die LED-Ausgaberegister des PCA 9532 für dieses Segment einzutragen (für PWM-Betrieb). Diese Masken sind dann im Programm an der entsprechenden Stelle in die Zielregister im PCA 9532 zu laden.

### Die Hauptschritt看ette

Nachdem die wichtigsten Initialisierungen beschrieben sind, können nun die Voreinstellungen für den Ablauf der Hauptschritt看ette in der „main function“ des Programms im Seeeuino erfolgen. Um einen Startzustand für den gesamten Ablauf herzustellen, sind nachfolgende Schritte zu empfehlen:

1. Alle LEDs der Fächer aus: PWM-Register des TLC59116 auf der I<sup>2</sup>C-LED-Baugruppe mit einem PWM-Wert voreinstellen (z B. für 50 %: 0x7F), Ausgaberegister 14 bis 17 des TLC59116 mit 0x0000 laden alle Leds aus
2. Die Ansteuerung der Transistoren für die Versorgung der Koppler CNY 70 über die Ausgabebaugruppe aus Bild 4 auf „Aus“ einstellen: PCF 8574A mit 0xFF laden
3. Die Struktur uc8\_tueren[8] löschen, kein Eingang eingeschaltet
4. An die Eingangsbaugruppen in die Outputregister jeweils 0xFFFF schreiben, dies setzt die Ausgänge auf „high“; um definierte Eingangspegel zu erhalten, wird das Setzen der Ausgänge auf „high“ vor dem ersten Benutzen als Eingänge laut Datenblatt des Herstellers empfohlen [10]
5. An die Anzeigebaugruppe als Startwert „00“ schreiben – ist natürlich individuell anzupassen
6. Die Freigabe erteilen für den Start des MP3-Players, Start automatisch, nachdem die „richtige“ Tür geöffnet wurde
7. Freigabe für die akustische Warnung, wenn eine falsche Tür geöffnet wurde
8. Das Einschalten der Drehscheibe mit dem kleinen Pyramidenaufbau (beim Start des Titels) freigeben



## 9. Das Einschalten der Beleuchtung des Drehtellers (beim Start des Titels) freigeben

Sind alle LEDs der Fächer AUS, wird als letzte Vorbereitung noch die „Betriebsbereitschaft“ des Arduino Uno abgefragt. Dazu wird das Zeichen „S“ („S“ wie Status) per UART vom Seeeduino an den Arduino Uno gesendet. Dieser antwortet zur Bestätigung mit „A“ („A“ wie „acknowledge“). Nun steht dem Start des Ablaufes in der Main Function nichts mehr im Wege. In der Main Function sind die nachstehend beschriebenen Schritte in einer State Machine in einer sich ständig wiederholenden Schleife abzuarbeiten:

1. Auslesen der 24 Reflexlichtschranken und Ermittlung der geöffneten Tür(en) sowie Abfrage und Auswertung der Befehlseingabe von der Fernbedienungsbaugruppe IRE 8
2. Abfrage des aktuellen Datums vom Arduino (genaugenommen von der hier aufgesteckten RTC-Uhr) und die Ausgabe an die Anzeigebaugruppe; diese Abfrage wird mit Kommando 2 der UART-Kommunikation vorgenommen; die Ausgabe erfolgt nur, wenn das Datum sich geändert hat; ab der erstmaligen Anzeige des aktuellen Tages ist der Kalender quasi „betriebsbereit“
3. Prüfen, ob die richtige Tür geöffnet wurde; wenn nein, weiter mit Schritt 4; wenn ja, den Startbefehl an den MP3-Player mit UART-Kommando übergeben mit der zum aktuellen Tag zugehörigen Titelnummer; Ausgabe der Ansteueremaske an die I<sup>2</sup>C-LED-Baugruppen zur Beleuchtung des geöffneten Faches; bei entsprechender Freigabe wird der Startbefehl an das Motor-Shield zum Antrieb der Drehscheibe gegeben und deren Beleuchtung eingeschaltet
4. Prüfen, ob im Schritt 3 ein Titel gestartet wurde; wenn kein Titel gestartet wurde, weiter mit Schritt 5; wenn ja, erfolgt hier die Abfrage des Endes des gestarteten Titels zum Beenden des Ablaufes und zur Freigabe eines erneuten Starts; ist der Titel beendet, wird der Startbefehl an das Motor-Shield zurückgesetzt, die Beleuchtung des Drehtellers wird abgeschaltet; spätestens hier wird auch die Beleuchtung des Faches ausgeschaltet, ansonsten schon beim ersten Schließen der Tür nach dem Start des Titels
5. Senden einer in Schritt 1 über die Fernbedienung eventuell empfangenen neuen Lautstärkevorgabe an die Audio-Baugruppe (über den Arduino Uno), auch während der Wiedergabe des Titels möglich; die Lautstärke wird mit den „+“- und „-“-Tasten der Fernbedienung zur Erhöhung oder Verminderung vorgegeben, die Vorgabe der Schrittweite dazu erfolgt in der Funktion zur Bearbeitung der IRE-8-Baugruppe; ein aktuell eingestellter Wert wird für 3 Sekunden an die 7-Segment-Anzeige ausgegeben; Ende der aktuellen Schleife und erneute Bearbeitung der Schleife ab Schritt 1

Zur Ausführung dieser Schritte ist es sinnvoll, die Aufgaben weiter zu unterteilen. Es sei an dieser Stelle angemerkt, dass der Arduino Uno während der Wiedergabe eines Titels bei Einsatz der mit der Audio-Baugruppe ASA1 mitgelieferten Software (Projekt ASA1.ino, [16]) weitgehend ausgelastet ist. Der Ausgang des MP3-Playes DREQ zur Meldung „Ausgabepuffer des MP3-Players leer“ wird ständig „abgepollt“, um die nächsten 32 Byte der Wiedergabedatei zu senden. Daher ist nur die Weitergabe einer aktualisierten Lautstärke (über UART empfangen) während des Abspielens des Titels in der Abfrageschleife für DREQ vorgesehen. Um dies zu ermöglichen, wurde die Ausgabefunktion VS10xx.cpp aus der zur ASA1 mitgelieferten Software angepasst. Auch kann in dieser Zeit die RTC-Uhr nicht abgefragt werden, ohne die Wiedergabe „kurz ins Stottern“ geraten zu lassen. Diese Feststellung und die Tatsache, dass der Ablauf zur Abfrage der Eingänge und die Ansteuerung der Ausgänge in der Zeit zwischen 2 auszugebenden 32 Byte an den MP3-Player nicht zu handeln ist, hat auch zum Einsatz einer zweiten Steuerung in Form des Seeeduino geführt.

## Türkontakte auslesen

Von den bereits erwähnten Teilaufgaben soll als Erstes die Funktion zum Auslesen der 24 Türkontakte näher beschrieben werden. Der Aufbau der Funktion ist natürlich abhängig von der Art der Überwachung der Türen. In der oben erwähnten Kalendervariante 1 wurden durch einen kräftigen Neodyn-Magneten, der in der Tür eingeklebt ist, 2 metallische Federkontakte geschlossen. In Variante 2 wurden Mikroschalter mit Sprungkontakten eingesetzt. Beide Varianten erzeugen Prellen, welches durch wiederholtes Abfragen abgefangen werden muss. Zur Verbesserung der Zuverlässigkeit der Türüberwachung fiel die Wahl in Variante 3 auf Reflexlichtschranken.

Bevor die Eingänge gelesen werden können, sind die 4 Transistoren zur Versorgung der 24 Infrarotdioden der CNY 70 durchzusteuern. Dazu ist in einem ersten Schritt das Byte 0bxxxx1111 an die Ausgangsbaugruppe BG3 auszugeben (x steht für anderweitig nutzbare Ausgänge). Anschließend sind die 2 Eingangsbaugruppen einzulesen und die 4 Byte in der Struktur uc8\_tueren[0...3] abzuspeichern. An die Ausgangsbaugruppe BG3 ist nun das Byte 0bxxxx0000 auszugeben und damit die 4 Transistoren wieder auszuschalten. Da entsprechend dem Stromlaufplan auch der Beeper für den Warnton an Bit 4 der Ausgabebaugruppe angeschlossen ist, ist diese Ansteuerung immer mit einzubeziehen. Um möglichst alle „Versuche“ zum Öffnen einer Tür zu erfassen, muss die Abfrage der Türen aus Schritt 1 in jeder Schleife des Hauptprogramms erfolgen. Wer will, kann auch die Interrupts der PCF 8575, die an INT 0 angeschlossen sind, auswerten.

In einem zweiten Schritt sind die 3 ersten eingelesenen Byte in der Struktur uc8\_tueren[0...2] zu untersuchen, ob offene Türen registriert wurden (Bit = 1). Im vierten Byte stehen die eingelesenen Schaltausgänge der Fernbedienungsbaugruppe. Zum Durchforsten der Strukturen sind zwei „for“-Schleifen ineinander verschachtelt. Die äußere Schleife indiziert durch die Laufvariable i die Byte 0 bis 2 der Struktur uc8\_tueren[i], die innere Schleife indiziert durch die Laufvariable j die Bit 0 bis 7 der 3 Byte der Struktur uc8\_tueren[i]. Diese beiden Indizes dienen auch der Adressierung der Struktur sc\_TuerNummer[i].Nummer[j] für den dieser Tür zugeordneten Tag, der Struktur sc\_FileNummer[i].file[j] zur Indizierung der dieser Tür zugeordneten Titelnummer sowie der Struktur sc\_TuerLed[i].Leds[j] zur Indizierung der Ausgabemaske zur Ansteuerung der LED dieses Faches.

Wird in einem solchen Durchlauf erkannt, dass die richtige Tür (sprich Türnummer und aktuelles Datum stimmen überein) geöffnet wurde, wird ein Merker gesetzt und die Titelnummer sowie die Maske für die LED des Faches ausgelesen und an die aufrufende Main Function zurückgegeben. In der State Machine der Main Function in Schritt 3 wird bei gesetztem Merker der Startbefehl für die Wiedergabe und die dazu gewünschte File-Nummer aus sc\_FileNummer[i].file[j] über die UART-Kommunikation mit Kommando 3 an den Arduino Uno gesendet.

In der zweiten LED-Baugruppe ist am Ende der





Umspeicherung der Maske noch die Ansteuerung der LED-Ausgänge 9 bis 16 mit zu übernehmen. Hier sind an LED 9 eine blaue LED zur Anzeige der gestarteten Wiedergabe des MP3-Files und an LED 10 und 11 LEDs zur Beleuchtung des Drehtellers angeschlossen.

Da in Byte uc8\_tueren[3] die eingelesenen 8 Eingänge des Infrarot-Fernbedienungsempfängers IRE 8 [0] abgelegt sind, kann hier die Auswertung der Schaltausgänge angeschlossen werden. Auf der Baugruppe IRE8 wurde die Reaktion auf Betätigung einer Taste der Fernbedienung auf „Ausgangssignal Ein solange die Taste betätigt wird“ eingestellt. Die Auswertung der betätigten Taste auf der Fernbedienung und die Ausführung des damit verbundenen Befehls sind im Anhang „Softwarebeispiele“ im Punkt 2 beispielhaft angegeben.

### Kommunikation zwischen Arduino Uno und Seeduino

Als nächste Funktion soll die Kommunikation zwischen dem Arduino Uno und dem Seeduino besprochen werden. Die Funktion zum Senden von Kommandos an den Arduino ist im Anhang [20] Punkt 3 als Beispiel angegeben. Folgende Kommandos sollen an den Arduino gesendet werden:

- **Kommando 1, Abfrage „Status Arduino“:**  
Es wird das Zeichen „S“ gesendet, als Antwort wird das Zeichen „A“ erwartet; diese Antwort wird als korrekte Initialisierung des Arduino angesehen
- **Kommando 2, Abfrage des aktuellen Datums:**  
Es wird das Zeichen „T“ gesendet, als Antwort wird die Zeichenkette „=T“ und anschließend der aktuelle Tag als ASCII-Zeichen erwartet
- **Kommando 3, Startbefehl für den MP3-Player:**  
Es ist ein Adventslied mit der übergebenen Nummer zu starten: es wird das Zeichen „P“, gefolgt von der Titelnummer als ASCII-Zeichen, gesendet; es wird keine Antwort erwartet

- **Kommando 4, Senden einer aktualisierten Lautstärke:**  
Wenn zuvor durch Betätigung der „+“- oder „-“-Taste der Fernbedienung eine neue Lautstärkeänderung vorgegeben wurde, wird die zuletzt gesendete Lautstärke unter Überwachung einer oberen und unteren Grenze (max. Lautstärke ist = 0, die minimale Lautstärke ist 255) um eine festgelegte Schrittweite erhöht oder vermindert und als ASCII-Zeichen an den Arduino gesendet, es wird keine Antwort erwartet. Im Arduino-Sketch ASA1\_V3.ino ist in der Initialisierungsfunktion eine mittlere Lautstärke von 70 für beide Kanäle vorgegeben (Lautstärke = 80 → sehr leise). Dieses Kommando wird bei Bedarf in Schritt 4 der Main Function aufgerufen

Zur Auswertung der per Interrupt empfangenen Antworten vom Arduino ist der Inhalt des UART-Empfangspuffers zu untersuchen. Der Auswertefunktion wird dazu in Variable RECEIVE\_DATE übergeben, welches Kommando zuvor gesendet wurde. Daraus erkennt die Funktion, ob und wenn ja welche Antwort zu erwarten ist. Beim Empfang des ersten Zeichens im UART-Puffer wird in der Empfangs-Interruptserviceroutine (ISR) der UART-Schnittstelle ein Merker gesetzt. Die Main Function fragt diesen Merker zyklisch in Schritt 4 ab. Nur wenn der Merker gesetzt ist, wird die Auswertefunktion aufgerufen.

Für Kommando 1 und 2 kann unmittelbar auf den vollständigen Empfang der Antwort gewartet werden. Nach Senden des Kommandos 3 zum Start des MP3-Files wird in der Main Function in jedem Programmdurchlauf der Empfangsmerker überprüft, um das Ende des Abspielvorganges des MP3-Files zu erfassen. Ist das File abgespielt, sendet der Arduino die Zeichenfolge „=end“. In Schritt 4 der Main Function wird diese Antwort abgefragt, wenn in Schritt 3 der Titel tatsächlich gestartet wurde. Wenn der Empfangsmerker als gesetzt erkannt wurde, wird nach einer kurzen Wartezeit die über die UART-Interruptserviceroutine empfangene Nachricht komplett in das array sc\_UartReceive[] umgespeichert.

Damit wird sichergestellt, dass alle Zeichen empfangen worden sind, auch wenn der Empfang gerade erst gestartet worden ist. Nach erfolgter Umspeicherung wird der Empfangsmerker wieder gelöscht.

Ist die Nachricht vom Beenden des Titels korrekt empfangen worden, wird die Beleuchtung des Faches abgeschaltet, der Startmerker für den MP3-Player rückgesetzt, der Drehteller gestoppt sowie seine Beleuchtung abgeschaltet. Damit ist ein kompletter Vorgang vom Öffnen der Tür bis zum Ende der Wiedergabe abgeschlossen. **ELV**

### Machen Sie mit!

Jede veröffentlichte Anwendung wird mit einem Warengutschein in Höhe von 200 Euro belohnt.



### Wir wollen es wissen – Ihre Anwendungen und Applikationen!

Welche eigenen kreativen Anwendungen und Applikationen haben Sie mit den ELV-Haus-technik-Systemen, aber auch mit anderen Produkten und Bausätzen realisiert – ob mit Standard-Bausteinen oder eingebunden in eigene Applikationen? Alles, was nicht gegen Gesetze oder z. B. VDE-Vorschriften verstößt, ist interessant. Denn viele Applikationen verhalfen sicher anderen zum Aha-Erlebnis und zur eigenen Lösung.

Schreiben Sie uns, fotografieren Sie Ihre Applikation, berichten Sie von Ihren Erfahrungen und Lösungen. Die interessantesten Anwendungen werden redaktionell bearbeitet und im ELVjournal mit Nennung des Namens vorgestellt.



**Per E-Mail**  
leserwettbewerb@elv.de



**Per Post**  
ELV Elektronik AG, Leserwettbewerb, 26787 Leer

Die Auswahl der Veröffentlichungen wird allein durch die ELV-Redaktion ausschließlich nach Originalität, praktischem Nutzen und realisierter bzw. dokumentierter Ausführung vorgenommen, es besteht kein Anspruch auf Veröffentlichung, auch bei themengleichen Lösungen. **Der Rechtsweg ist ausgeschlossen.** Für Ansprüche Dritter, Beschädigung und Verlust der Einsendungen wird keine Haftung übernommen. Alle Rechte an Fotos, Unterlagen usw. müssen beim Einsender liegen. Die eingesandten Unterlagen und Aufnahmen verbleiben bei der ELV Elektronik AG und können von dieser für Veröffentlichungen und zu Werbezwecken genutzt werden.



## Weitere Infos:

- [1] [www.elv.de](http://www.elv.de) ELV Audio Shield für Arduino ASA1; Best.-Nr. J6-10 59 22
- [2] [www.4dsystems.com.au](http://www.4dsystems.com.au) Link zum Hersteller des SOMO 14D
- [3] [www.elv.de](http://www.elv.de) Infrarot-Fernbedienungs-Baugruppe IRE 8, 8 Kanäle, Best.-Nr. J6-10 46 74
- [4] [www.elv.de](http://www.elv.de) MP3 Soundmodul MSM 2, Best.-Nr. J6-09 28 53
- [5] [www.arduino.cc](http://www.arduino.cc) Homepage des Arduino Uno
- [6] [www.elv.de](http://www.elv.de) I<sup>2</sup>C-Realtime-Clock I2C-RTC, Best.-Nr. J6-10 34 13
- [7] [www.seeedstudio.com](http://www.seeedstudio.com) Informationen über den Seeeduino v3.0 über die wiki page
- [8] [www.watterott.com](http://www.watterott.com) Link zum Lieferanten der Eingangsbaugruppe 1 und 2
- [9] [www.exp-tech.de](http://www.exp-tech.de) Link zum Distributor von Getriebemotor und Motor Driver Carrier DRV8835
- [10] [www.nxp.com/documents/data\\_sheet/PCF\\_8575.pdf](http://www.nxp.com/documents/data_sheet/PCF_8575.pdf)
- [11] [www.elv.de](http://www.elv.de) LED-I<sup>2</sup>C-Steuertreiber, 16 Kanäle, Best.-Nr. J6-09 83 77
- [12] [www.nxp.com/documents/data\\_sheet/PCA\\_9532.pdf](http://www.nxp.com/documents/data_sheet/PCA_9532.pdf)
- [13] [www.elv.de](http://www.elv.de) USB-I<sup>2</sup>C-Interface-Baugruppe; Best.-Nr. J6-08 41 23
- [14] [www.pololu.com](http://www.pololu.com) Link zur Motorsteuerung sowie zum Datasheet vom DRV8835
- [15] [www.elv.de](http://www.elv.de): Webcode #1334 Link zum Onlinebeitrag
- [16] Arduino-Sketch in der Version ASA1\_V3 (angepasste Version von ELV) s. a. [1]
- [17] [www.hpinfofotech.ro](http://www.hpinfofotech.ro) Demoversion CodeVisionAVR V3.12 Evaluation
- [18] [www.atmel.com](http://www.atmel.com) AVR Studio 6.2 und Datenblatt des ATmega 328P
- [19] [www.der-hammer.info/terminal](http://www.der-hammer.info/terminal) HTerm
- [20] [www.ti.com/product/tlc59116](http://www.ti.com/product/tlc59116) Datenblatt TLC 59116



# Anhang: Softwarebeispiele

## 1. Deklarationen und Definitionen von Variablen und Strukturen:

```
unsigned char uc8_tueren[4];
struct FileNumber{
    unsigned char file[8];
};
struct TuerNummer{
    unsigned char Nummer[8];
};
struct TuerLed{
    unsigned int Leds[8];
};
//in dieser Struktur sind die auf den Türen 1 bis 24 individuell vergebenen Türnummer
//einzutragen:
struct TuerNummer sc_TuerNummer[4] = {
    {1,2,3,4,5,6,7,8},
    {9,10,11,12, 13,14,15,16},
    {17,18,19,20,21,22,23,24},
    {25,26,27,28,29,30,31,32}
};
//in dieser Struktur sind die Titelnummern zugeordnet zu der Platzierung der Türnummer
//in Struktur scTuerNummer[] einzutragen:
struct FileNumber sc_FileNumber[4] = {
    {1,2,3,4,5,6,7,8},
    {9,10,11,12,13,14,15,16},
    {17,18,19,20,21,22,23,24}
};

//in dieser Struktur sind die Einschaltmasken für die LED's der Fächer 1 bis 24 einzutragen
//entsprechend ihrer tatsächlichen Verschaltung (es ist immer nur eine LED eingeschaltet):
struct TuerLed sc_TuerLed[4] = {
    {0x0002,0x0008,0x0020,0x0080,0x0200,0x0800,0x2000,0x8000},
    {0x0002,0x0008,0x0020,0x0080,0x0200,0x0800,0x2000,0x8000},
    {0x0002,0x0008,0x0020,0x0080,0x0200,0x0800,0x2000,0x8000},
    {0x0002,0x0008,0x0020,0x0080,0x0200,0x0800,0x2000,0x8000}
};

unsigned int sc_7SegDay[] = {0x1515,0x0014,0x4505,0x4115,0x5014,0x5111,0x5511,0x0015,0x5515,0x5115};
//wenn Bit 0 = Segment a, Bit 1 = Segment b usw.
```

## 2. Die Auswertung der betätigten Taste auf der Fernbedienung und die Ausführung des damit verbundenen Befehls:

```
//Auswertung der 8 Input-Bits;
//das Kommando entsprechend dem Tastendruck ermitteln
//das eingelesene Byte von der Fernbedienungsbaugruppe steht in Variable uc_Tasten
if(uc_Tasten & 0x80){
    //- Taste „1“: Buzzer enable/disablen
    b_enable_Beeb = !b_enable_Beeb;
    uc_Step = 0x00;
}
else if(uc_Tasten & 0x40){
    //noch kein Befehl hinterlegt
    uc_Step = 0x00;
}
else if(uc_Tasten & 0x20){
    // Taste „2“: Drehen des Tellers enable/disablen
    b_TellerEnable = !b_TellerEnable;
    uc_Step = 0x00;
}
else if(uc_Tasten & 0x10){
    // Taste „+ Volume“: Lautstärke +
    //neuen Wert für Lautstärke zur Ausgabe an den MP3 player aktuell errechnen
    //max. Lautstärke == 0x00 !
    if (uc_Laut >= 0x07) {
        uc_Laut -= 0x05; //Schrittweite zur Erhöhung der Lautstärke
    }
    //Merker setzen, neue Lautstärkeeinstellung ist noch auszugeben
    uc_PushLaut = 0x01;
}
else {
    uc_Laut = 0x00; //max. Lautstärke erreicht,
    b_DpDig2 = 0x01; //Dezimalpunkt rechts ein
    uc_LautMax = 0x01;
};
uc_Step = 0x00;
}
else if(uc_Tasten & 0x08){
    // Taste „- Volume“: Lautstärke -
    //min. Lautstärke ca. 80d, schon sehr leise
    if (uc_Laut <= 0x80) {
```



```

        uc_Laut += 0x05; //Schrittweite zur Verminderung der Lautstärke
        uc_PushLaut = 0x01; //Merker setzen, neue Lautstärkeeinstellung ist noch
// auszugeben
    }
    uc_Step = 0x00;
}
else if(uc_Tasten & 0x04){
    // Taste 6: noch kein Befehl hinterlegt
    uc_Step = 0x00;
}
else if(uc_Tasten & 0x02){
    // Taste 7: noch kein Befehl hinterlegt
    uc_Step = 0x00;
}

else if(uc_Tasten & 0x01){
    // noch frei: noch kein Befehl hinterlegt
    uc_Step = 0x00;
}
else if(uc_Tasten == 0x00){
    // Keine Taste betätigt;
    uc_Step = 0x00;
}
else if((uc_Tasten != 0x01)&&(uc_Tasten != 0x02)&&(uc_Tasten != 0x04)&&(uc_Tasten != 0x08)&&
(uc_Tasten != 0x10)&&(uc_Tasten != 0x20)&&(uc_Tasten != 0x40)&&(uc_Tasten != 0x80))
{
    //Fehler, kein gültiges Kommando empfangen:
    uc_Step = 0xFF;
}
}

```

### 3. Funktion zum Senden von Kommandos an den ARDUINO:

```

//--- Funktion zur Kommunikation mit dem ARDUINO
void UART_Communication (unsigned char STEP, unsigned char DATE )
{
    switch (STEP)
    {
        case 0x10:
            //Kommando 1: Abfrage des Status des ARDUINO UNO
            uc_TxD = ,S'; //zu sendendes Zeichen
            putchar(uc_TxD); //Sendefunktion
            break;

        case 0x20:
            //Kommando 3: MP3 player starten
            uc_TxD = ,P'; //Startsignal
            putchar(uc_TxD);
            uc_TxD = DATE; //file number an ARDUINO UNO übergeben
            putchar(uc_TxD);
            break;

        case 0x30:
            //Kommando 2: Anforderung an ARDUINO, den aktuellen Tag zu senden
            uc_TxD = ,T'; //
            putchar(uc_TxD);
            break;

        case 0x40:
            //Kommando 4: Ausgabe der aktualisierten Lautstärke für den MP3 player
            uc_TxD = DATE; //Lautstärke aus DATE übernehmen
            putchar(uc_TxD);
            break;
    }
}
//--- ende Deklaration Funktion UART_Communication();

```

### 4. Auswertung des Empfangs vom ARDUINO UNO:

```

//Empfang auswerten:
//wenn Startbereitschaft abgefragt wurde:
if(RECEIVE_DATE == 'S'){
    if (sc_UartReceive[0] == ,A') {
        uc_MP3_ready = 0x01; //Merker für „MP3 player betriebsbereit“ setzen
    }
}
//wenn der Tag abgefragt wurde:
if(RECEIVE_DATE == ,T'){
    if ((sc_UartReceive[0] == ,=')&&(sc_UartReceive[1] == ,T')) {
//empfangenen Tag in die Variablen date_TWI speichern
        if(sc_UartReceive[0] == 0){
            //Tag nur einstellig:
            str_GetDate [0] = sc_UartReceive[2]; //empfangenen Tag umkopieren
            ui_Copy = atoi(str_GetDate); } //und in INT-Wert wandeln

```



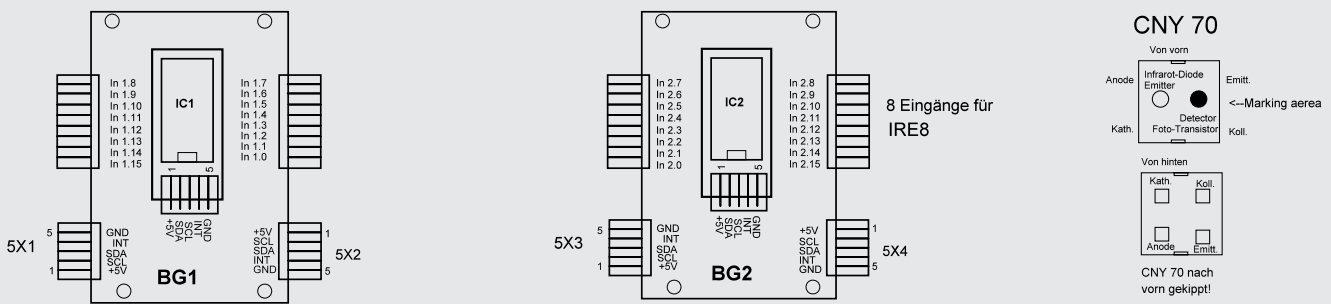
```

else {
  //Tag 2 stellig:

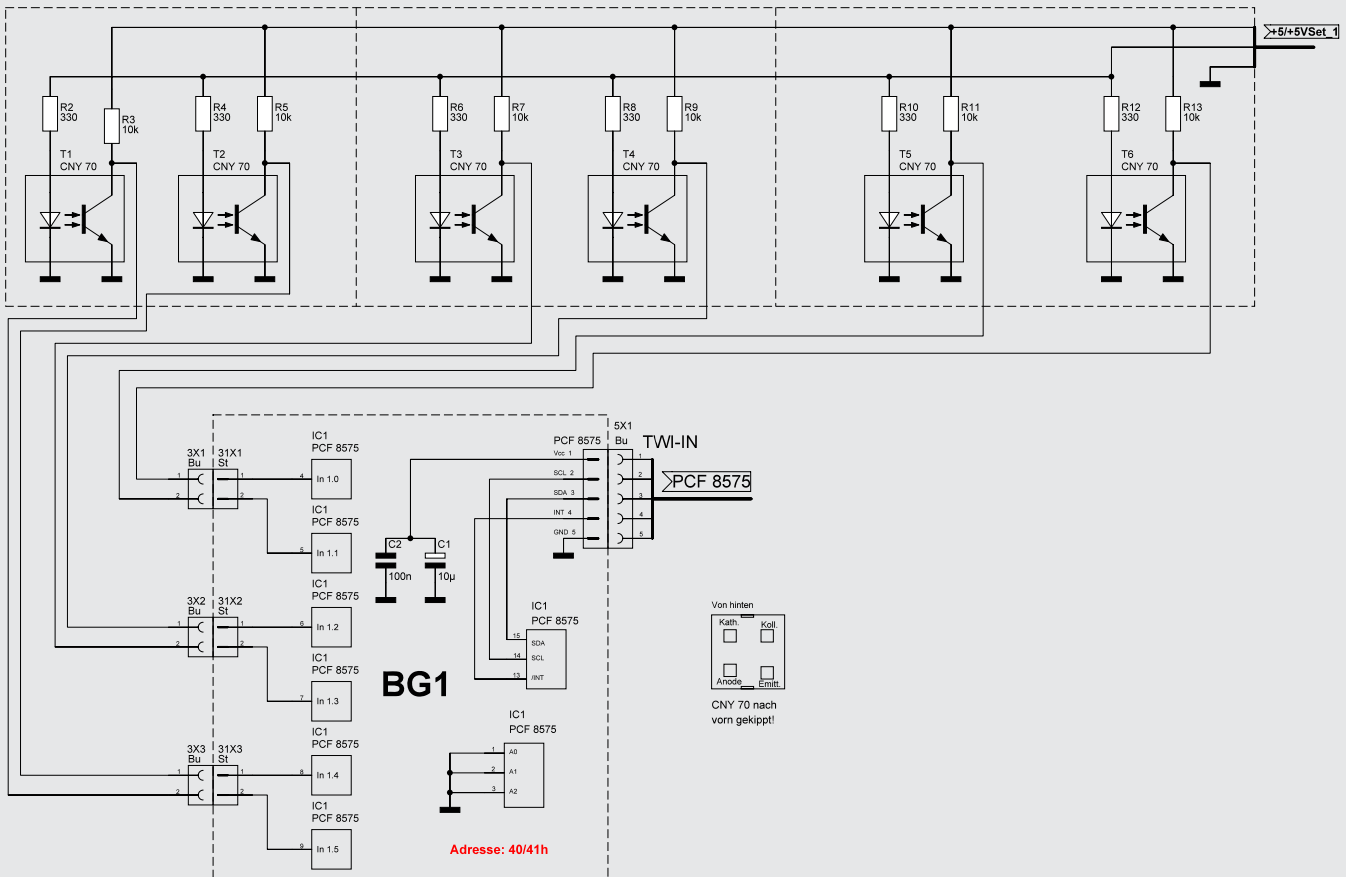
  str_GetDate [0] = sc_UartReceive[2]; //empfangenen Tag umkopieren
  str_GetDate [1] = sc_UartReceive[3]; //und in INT-Wert wandeln
  ui_Copy = atoi(str_GetDate);
}
};
//wenn das Titeldate abgefragt wurde:
if(RECEIVE_DATE == ,P')
{if ((sc_UartReceive[0] == ,=') && (sc_UartReceive[1] == ,e') && (sc_UartReceive[2] == ,n') && (sc_UartReceive[3] == ,d'))
{uc_MP3_Start = 0x00; //Titeldate empfangen, Startmarker löschen
uc_MP3_play = 0x00;
}
}
};

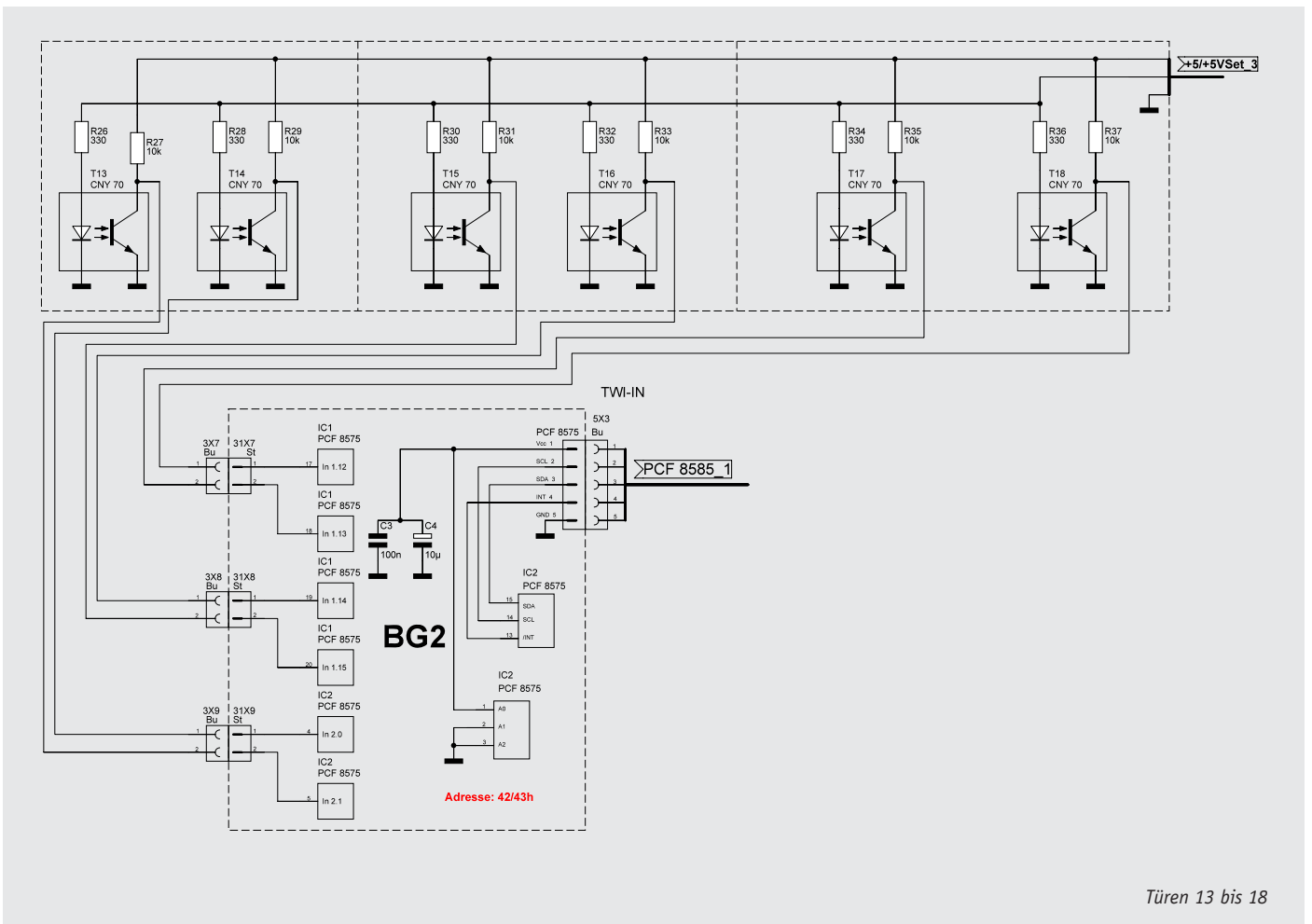
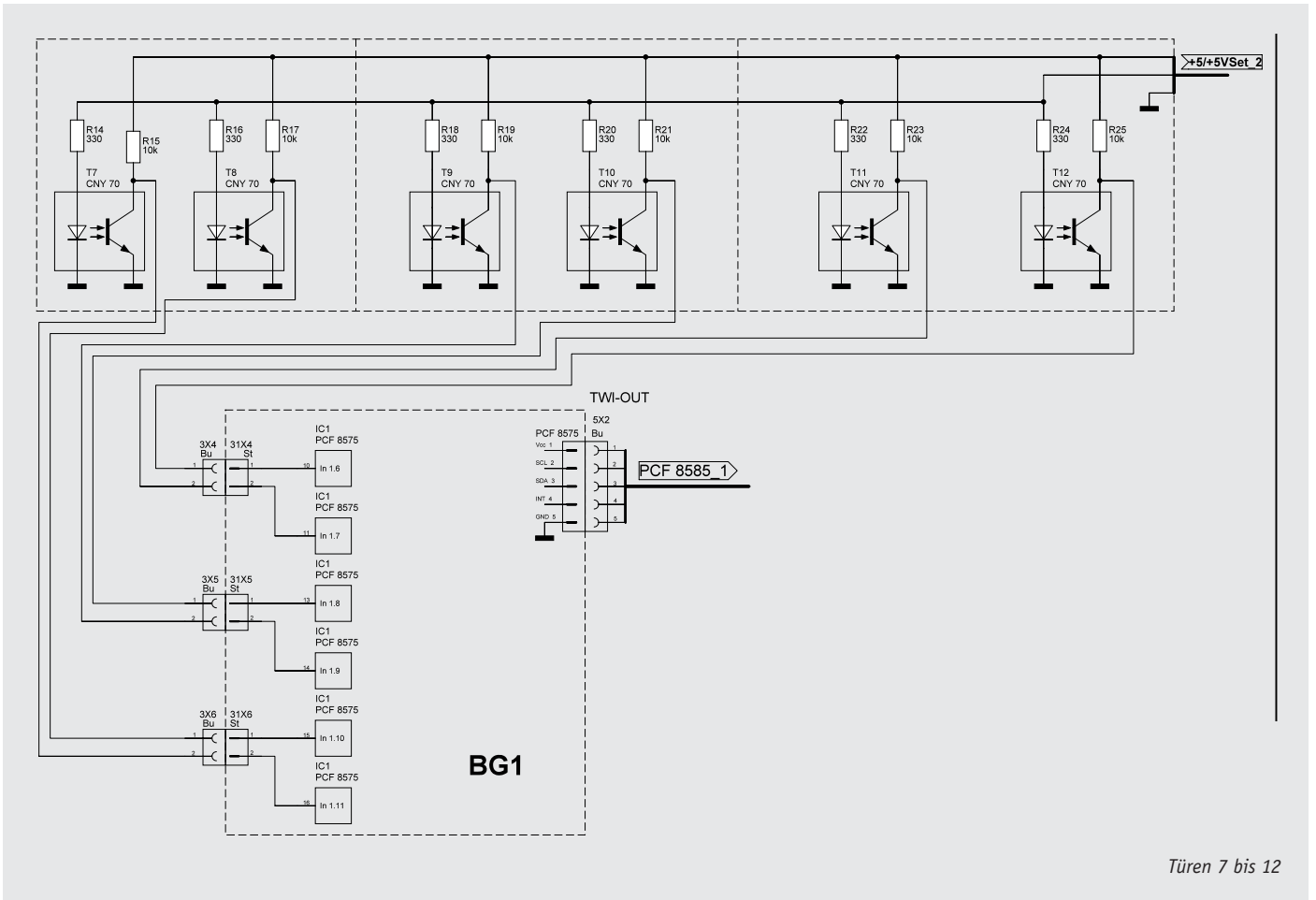
```

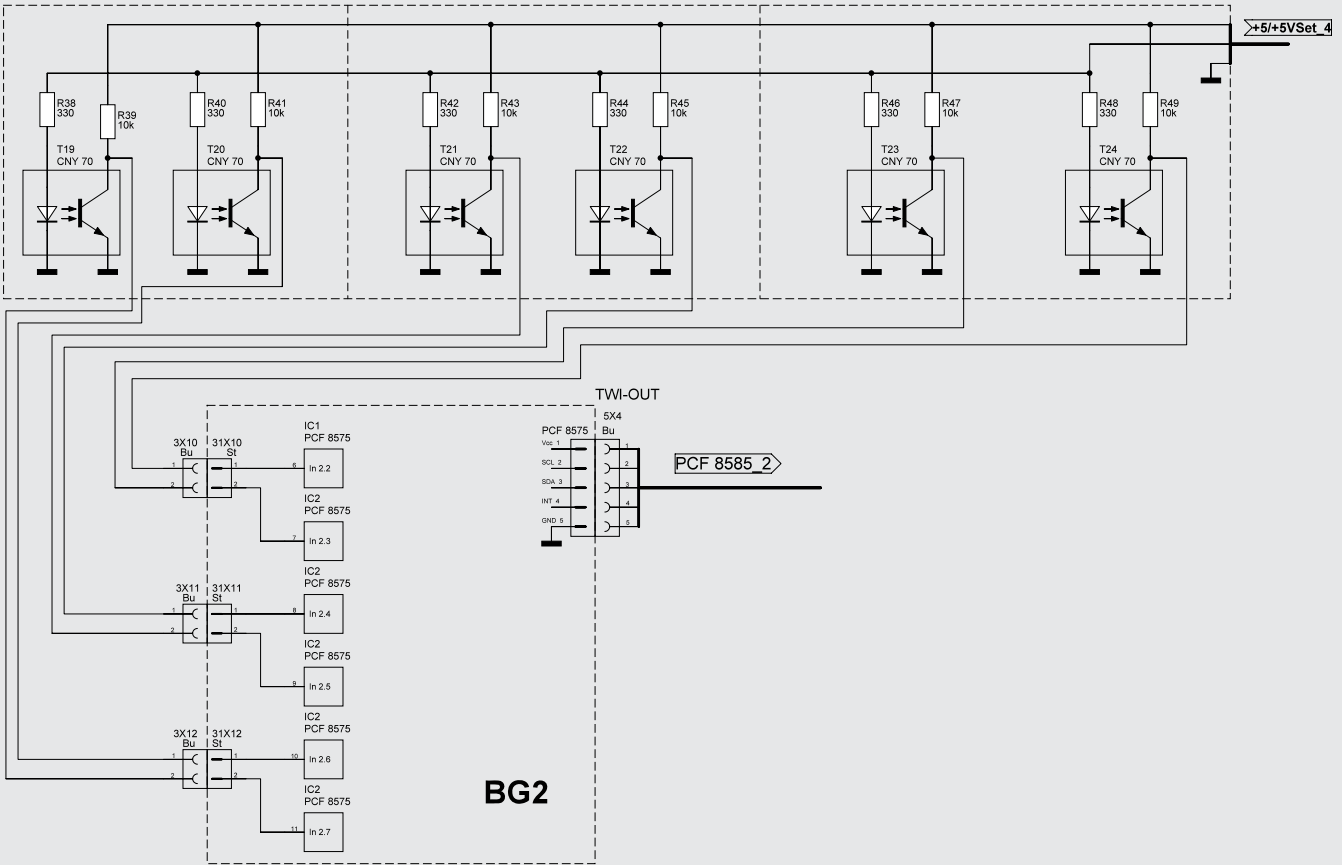
# Anhang: Teilschaltpläne



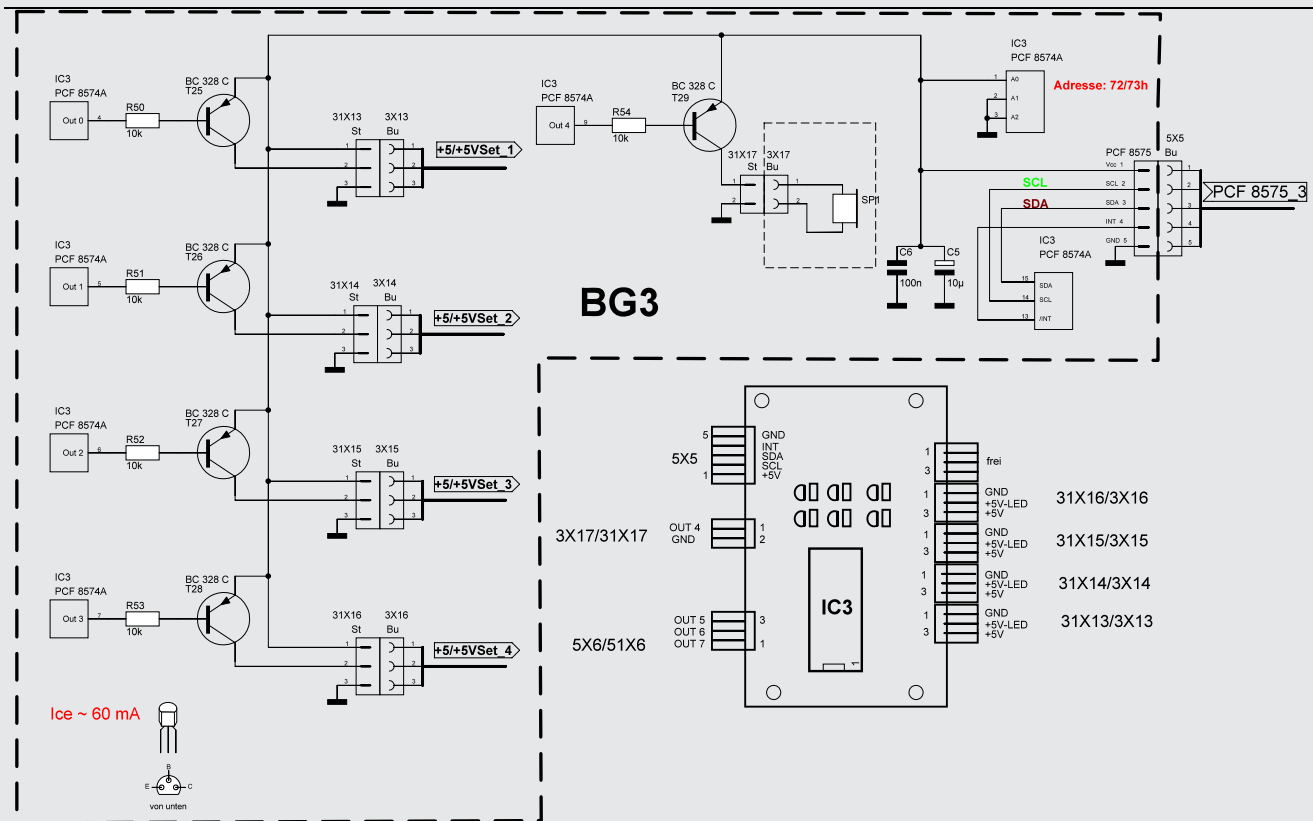
Anordnung und Anschlüsse der PCF 8575 auf BG1 und BG 2



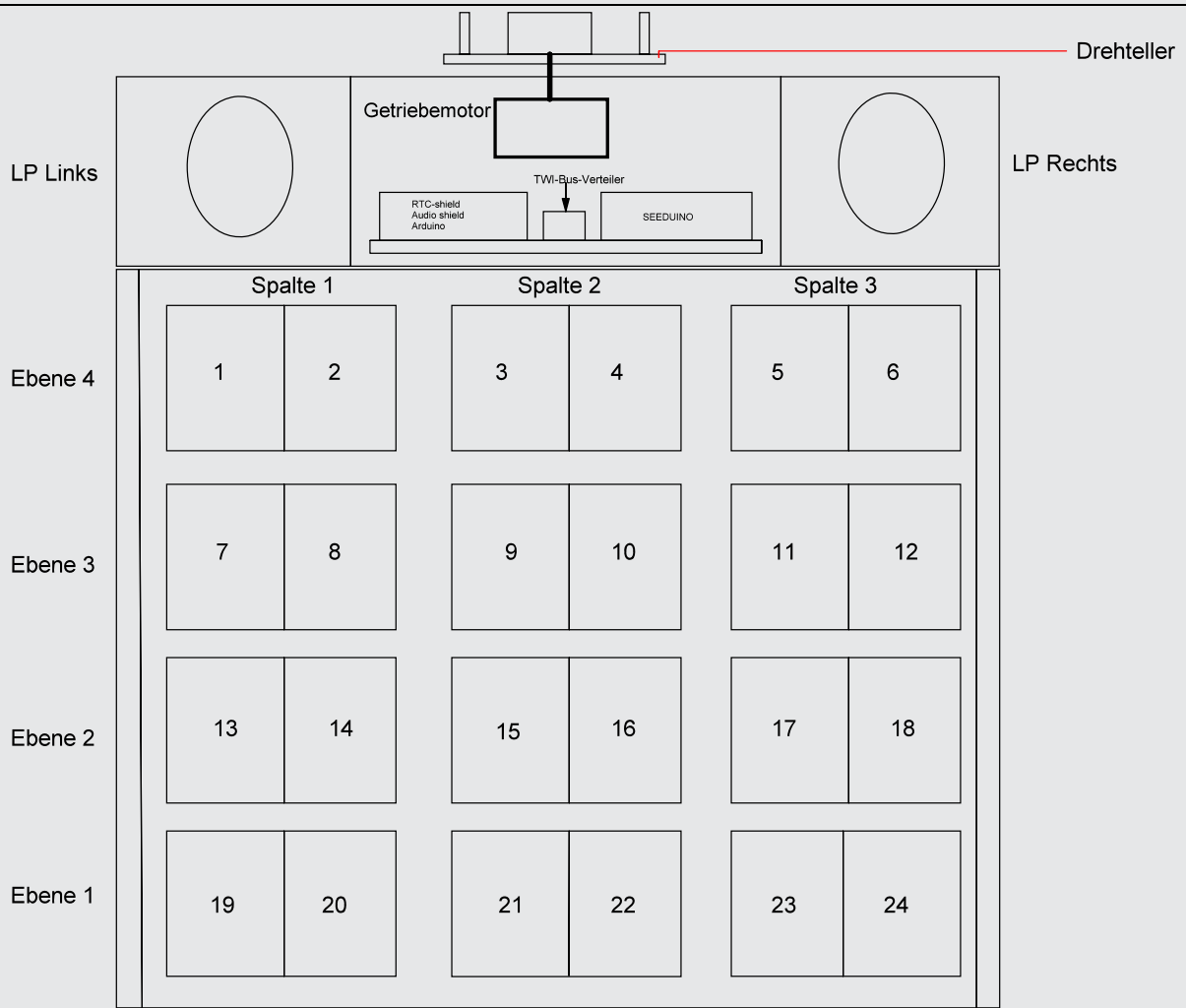




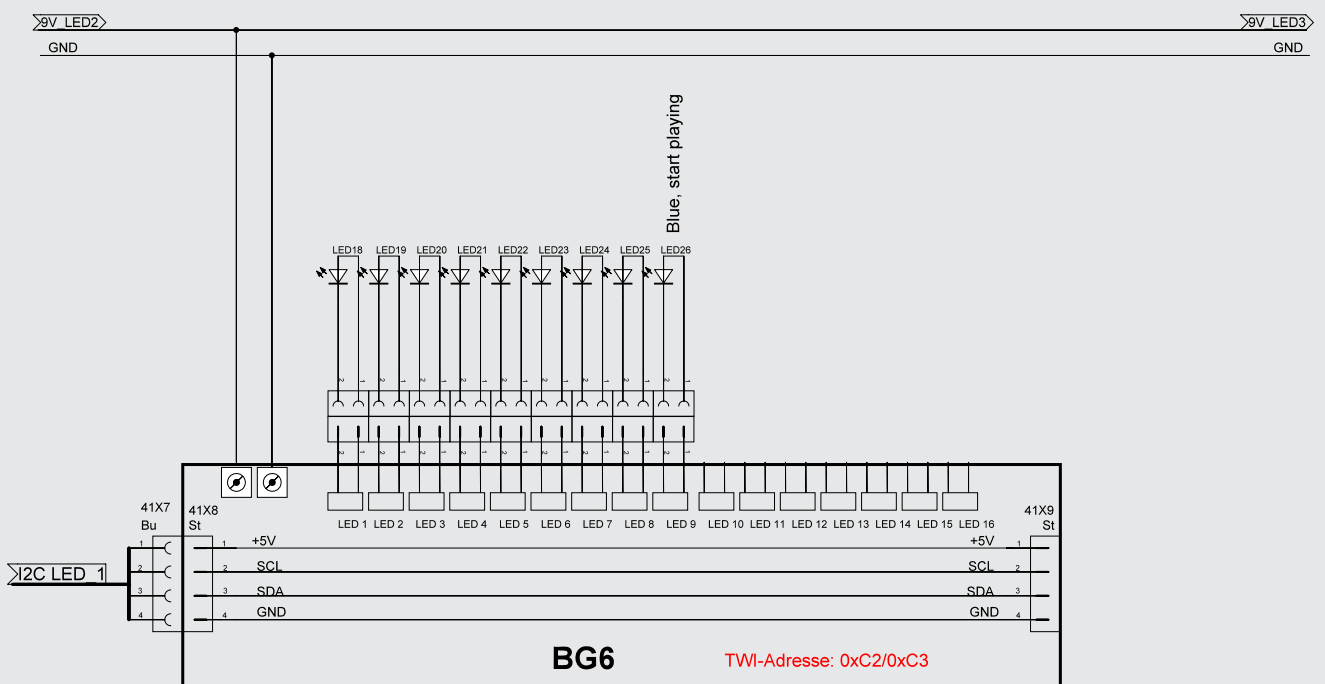
Türen 19 bis 24



Belegung und Beschriftung der 8 Ausgänge des PCF 8574A

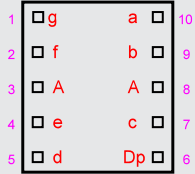
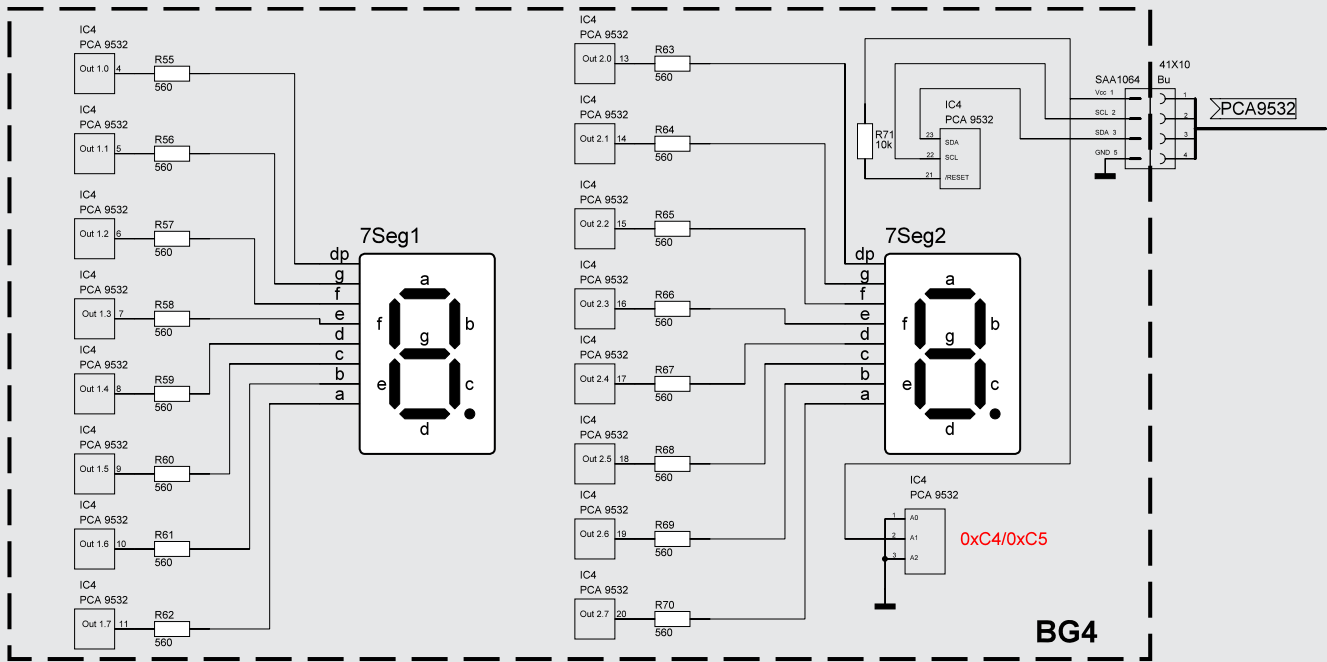


Die Anordnung der Fächer im Kalender



Die Beschaltung der Baugruppe 6 mit den LEDs 16 bis 31

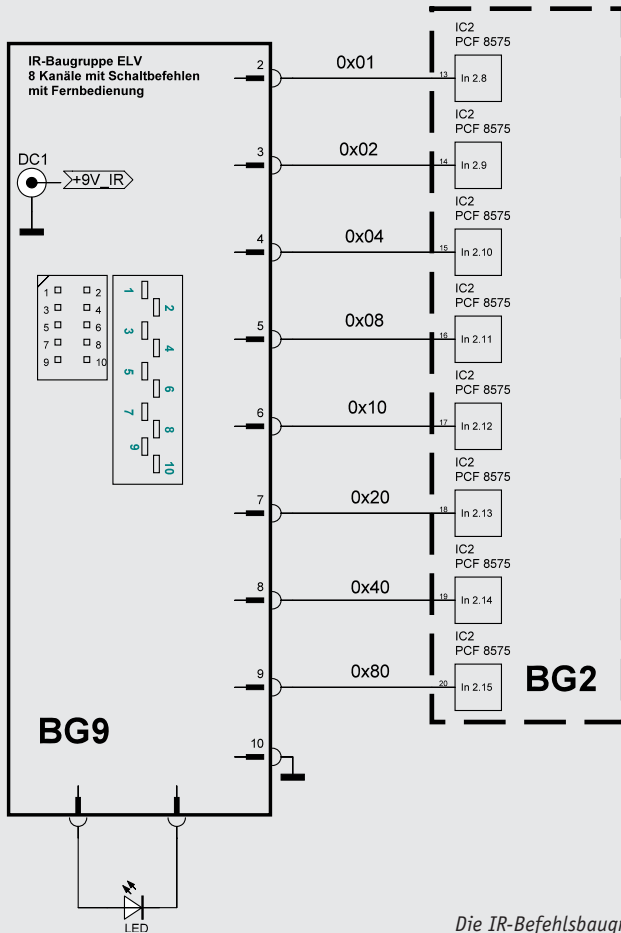




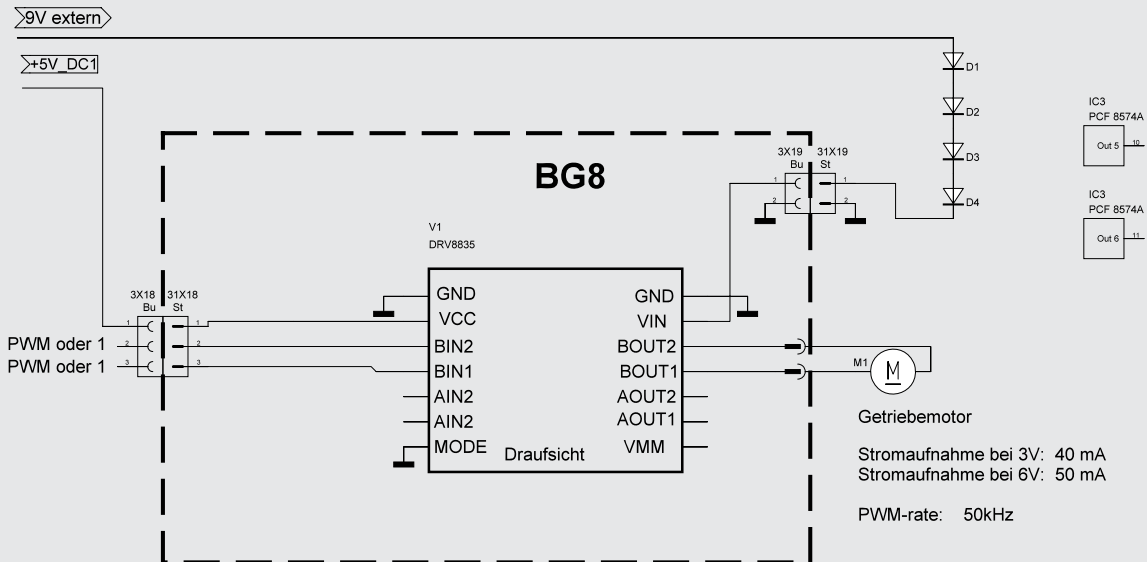
7-Segmantanzeige von oben

SA 39-11GWA  
RM 7,62

Die 7-Segmantanzeige mit PCA 9532



Die IR-Befehlsbaugruppe



Drive/coast or drive/brake operation with MODE=0 (IN/IN)

xIN1	xIN2	xOUT1	xOUT2	operating mode
0	0	OPEN	OPEN	coast (outputs off)
0	PWM	L	PWM	reverse/coast at speed PWM %
PWM	0	PWM	L	forward/coast at speed PWM %
PWM	1	L	PWM	reverse/brake at speed 100% - PWM %
1	PWM	PWM	L	forward/brake at speed 100% - PWM %
1	1	L	L	brake low (outputs shorted to ground)

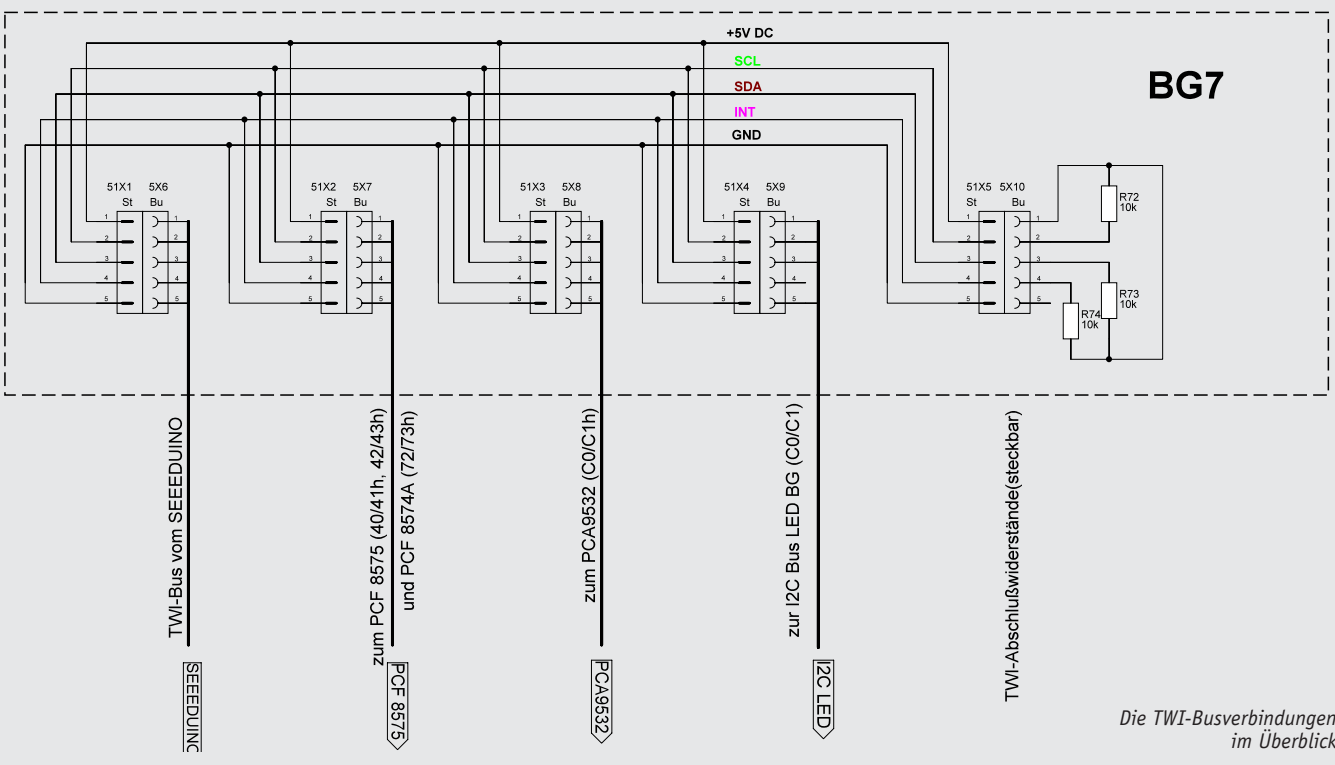
RECOMMENDED OPERATING CONDITIONS

T<sub>A</sub> = 25°C (unless otherwise noted)

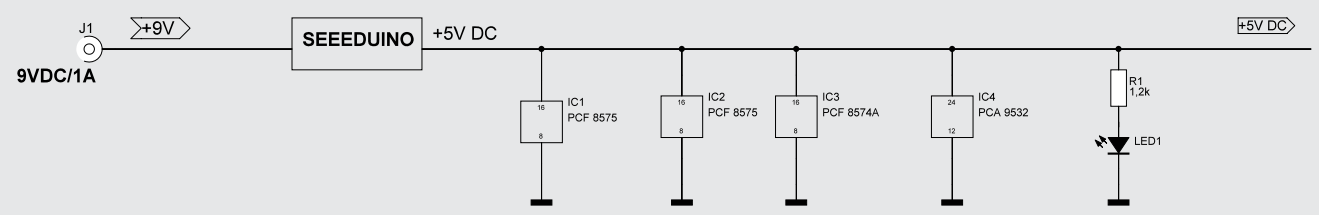
	MIN	NOM	MAX	UNIT
V <sub>CC</sub>	Device power supply voltage range	2	7	V
V <sub>IN</sub>	Motor power supply voltage range	2	11	V
I <sub>OUT</sub>	H-bridge output current <sup>(1)</sup>	0	1.5	A
f <sub>PWM</sub>	Externally applied PWM frequency	0	250	kHz
V <sub>IN</sub>	Logic level input voltage	0	V <sub>CC</sub>	V

(1) Power dissipation and thermal limits must be observed.

Die Ansteuerung des Getriebemotors für die Drehscheibe



Die TWI-Busverbindungen im Überblick



Stromversorgung/Adressierung der TWI-ICs

