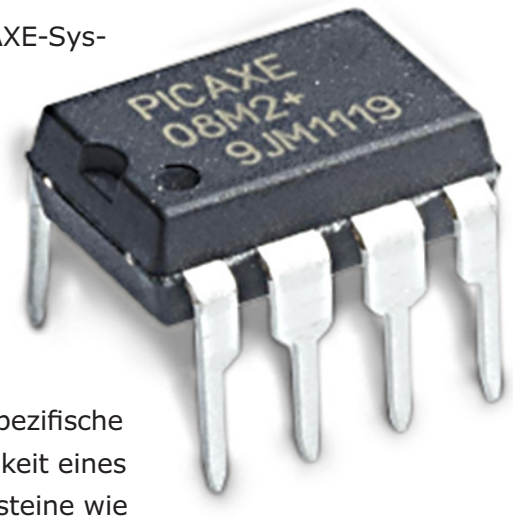
BASIC für PICs (2)

I/O-Schaltungen für PICAXE



Der erste Artikel dieser Serie beschäftigte sich mit dem PICAXE-System [1] und zeigte, wie man eine PICAXE-Programmierschaltung bauen, den Chip damit programmieren und (beispielsweise) eine Tasten/LED-Kombi steuern kann.
In dieser Folge geht es um Schaltungen mit gebräuchlichen Elektronikbauteilen, die zur Ein- und Ausgabe am PICAXE-Chip angeschlossen werden. Der Artikel soll Ihnen zeigen, welche Faktoren zu berücksichtigen sind, wenn Sie Bauteile für Ihr eigenes „Interfacing“ verwenden wollen.
Die folgenden Artikel decken dann fortgeschrittenere, chip-spezifische Schnittstellen-Funktionen von PICAXE ab. Die Leistungsfähigkeit eines PICAXE-Projekts kann durch spezielle ICs und Peripheriebausteine wie zusätzliche Speicher, Tastaturen, LCD-Displays oder sogar einen über eine serielle Verbindung angeschlossenen PC gesteigert werden.

In interrete veritas!

Im Internet steht eine Fülle von Informationen, für alle, die gerade erst anfangen, ihre eigenen auf Mikrocontrollern basierten Elektronik-Designs zu entwickeln, zur Verfügung. Leider ist es meist sehr schwer, zu den gezeigten konkreten Interfaceschaltungen ausreichende (theoretische) Hintergrundinformationen an gleicher Stelle zu bekommen, um zu verstehen, warum ausgerechnet dieses Bauteil mit diesem bestimmten Wert verwendet wurde. Dies macht es unmöglich, die Schaltungen eigenen Bedürfnissen anzupassen. Deshalb haben wir den hier gezeigten Beispielschaltungen etwas Theorie angefügt, damit Sie genau wissen, wie die richtigen Komponenten für Ihre Entwürfe zu wählen sind.

Startpunkt: Der Versuchsaufbau

Alle hier beschriebenen Experimente werden auf einem PICAXE 08M2 (Anschlussbelegung in **Bild 1**) durchgeführt, der an einem ATX-Power-Supply mit 5 V arbeitet. Die Beispielschaltungen sind lötfrei auf Steckboards aufgebaut. Die Programmierung der PICAXE-Chips erfolgt über ein USB-nach-Seriell-Kabel nach der Methode, die im vorherigen Artikel beschrieben wurde [2], mit der LinAXEpad Softwareversion 1.5.0 für (Arch) Linux. Um Verwechslungen zu vermeiden, entsprechen alle Pin-Nummern der Schaltungen den physikalischen Pin-Nummern des Chips (für alle Beispiele: Pin 3 ist Ausgang, Pin 4 ist Eingang). Pin-Nummern in Programmlistings verweisen auf die internen Pin-Namen gemäß

Von **Wouter Spruit**
(Niederlande)

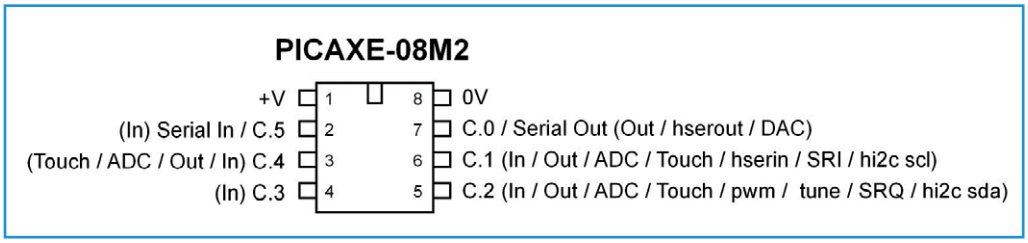


Bild 1.
Anschlussbelegung
PICAXE 08M2.

Bild 1. Wie man einen PICAXE-Chip programmiert, können Sie im vorherigen Artikel [2] oder im PICAXE-Handbuch auf der Website [3] nachlesen. PICAXE-Chips und verschiedene Zusatzschaltungen sind über den Revolution Education Webshop verfügbar [4].

Schütze deine Bauteile!

Bauteile haben Grenzwerte für Spannung und Strom, deren Überschreitung sie wahrscheinlich dauerhaft beschädigen dürfte. Strom, Spannung und Widerstand stehen nach dem Ohmschen Gesetz im Zusammenhang: Spannung (U) = Strom (I) * Widerstand (R). Der Strom durch ein Bauteil ist eine direkte Folge des Spannungspotentials und des Widerstands zwischen zwei Punkten. Bei einer festen Gleichstromquelle, zum Beispiel 5 V für einen PICAXE-Controller, kann der Strom begrenzt werden, wenn man einen Widerstand in Reihe hinzufügt. Sind mehrere Widerstände in Reihe geschaltet, steigt der Wert (R = R1+R2+R...+Rn), werden sie parallel geschaltet, wird der Wert mit jedem Widerstand niedriger (1/R = 1/R1+1/R2+...+1/Rn). Der Strom durch die Komponenten in Serie ist konstant, auch wenn sie unterschiedlichen Widerstand aufweisen. Nach dem Ohmschen Gesetz ist der Spannungsabfall über die einzelnen Bauteile proportional zu ihrem Widerstand. Jeder mit Interesse an Elektronik sollte das Ohmsche Gesetz im Schlaf herbeten können. Mehr Informationen darüber sind leicht im Internet zugänglich, zum Beispiel auf dieser Webseite [5]. Zum Lernen ist es nie zu spät! Wenn man folgendes versteht, macht dies die Berechnung von Widerständen in realen Schaltungen viel einfacher. Betrachten Sie die Spannungsteiler in **Bild 2**. Die Widerstände R1 und R2 sind in Reihe geschaltet sind, so dass der Strom von der Spannungsquelle zur Masse konstant ist. Jedoch unterscheidet sich der Spannungsabfall über R1 und R2 entsprechend ihrer Widerstandswerte. Weil das Ohmsche Gesetz für Potentialdifferenzen gilt, funktioniert es auch für den Zusammenhang zwischen Spannung, Strom und Widerstand in nur einem Teil der Schaltung. Die Spannung über R1+R2 beträgt +5 V, die Spannung über R1 ist dann +5 V*(R1/(R1+R2)) = 5 V*(5k/15k) = 1,67 V. Logischerweise fällt die verbleibende Spannung über R2 von 5 V - 1,67 V = 3,33 V = 5*(R2/(R1+R2)) ab. Dies gilt unbedingt für ohmsche Komponenten, aber nicht unbedingt für andere wie Dioden (einschließlich LEDs und Optokoppler-Eingänge) und Transistoren. Um den Widerstand zur Strombegrenzung bei einer LED zu berechnen, ermittelt man den Spannungsabfall über der LED (messen oder im Datenblatt nachsehen), anstatt zu versuchen, ihren ohmschen Widerstand mit einem Multimeter messen! Beispiel: Wenn eine rote LED mit einem Spannungsabfall von 1,8 V aus einer +5 V-Quelle mit einem Strom von 10 mA betrieben werden soll, berechnen wir den Strombegrenzungswiderstand wie folgt: Spannung über dem Widerstand = 5 V-1,8 V (LED-Spannungsabfall) = 3,2 V. Jetzt verwenden wir U/I = R = 3,2 V/0,010 A = 320 Ω. Der Widerstand (in der E6-Serie), der diesem Wert am nächstem kommt, ist 330 Ω.

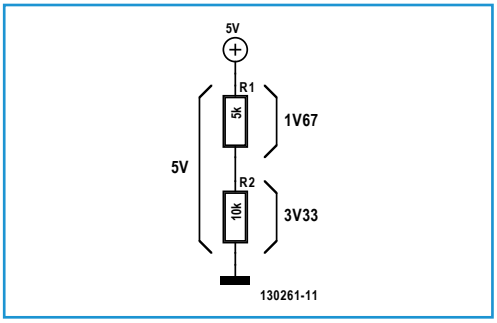


Bild 2.
Beispiel für einen Spannungsteiler.

Das Prinzip des Spannungsteilers kann an Mikrocontroller-Eingängen sehr nützlich sein. Wenn man R1 oder R2 aus Bild 2 durch einen variablen Widerstand ersetzt, der zum Beispiel die Temperatur oder die Lichtintensität messen kann, steht eine proportionale Spannung zwischen R1 und R2 zur Verfügung, die man mit dem Mikrocontroller-A/D-Wandler abgreifen könnte.

Da es eine Grenze für den Strom gibt, der in den Chip hinein (sink) oder aus den Chip heraus (source) fließen kann, ist es überaus wichtig, diesen Strom in jeder der (Schnittstellen-)Schaltungen zu begrenzen. In vielen Fällen benötigen Peripheriebauteile allerdings mehr Strom, als der Chip liefern beziehungsweise aufnehmen kann.

Der Transistor

Es würde den Rahmen dieses Artikels bei weitem sprengen, auf die Besonderheiten aller Transistor-Typen einzugehen. Wir zeigen am Beispiel des bekannten und leicht erhältlichen NPN-Transistors BC547B (oder Äquivalent), wie man die theoretische Begrenzung des

Stroms durch den Mikrocontroller umgeht. Die Idee ist, dass ein schaltender Transistor verwendet wird, der in der Lage ist, mit dem für die Anwendung notwendigen Strom problemlos fertig zu werden.
Ein NPN-Transistor reguliert den Strom, der von seinem Kollektor zum Emitter fließt, in Abhängigkeit von der Spannung an seiner Basis. Ein kleiner Strom zur Basis führt zu einem großen Strom durch die Kollektor-/Emitter-Strecke, so dass der Transistor als Verstärker arbeitet, bis der Transistor überhaupt keinen Strom mehr begrenzt (gesättigt ist). Ein PNP-Transistor invertiert übrigens den Schaltvorgang, aber das soll uns hier nicht weiter beunruhigen.
Bewaffnet mit diesem Wissen bauen wir eine Schaltung, um den Transistor als Schalter für eine LED zu testen: Der Strom aus einem PICAXE-Ausgang steuert den Transistor, der wiederum die LED schaltet. Der PICAXE-Ausgang muss nur noch den geringen Strom liefern, um den Transistor anzusteuern; der Strom für die LED wird in diesem Fall direkt vom Netzteil bereitgestellt.
Im Transistor-Beispiel wird die gleiche (Taster-)Schaltung wie im einleitenden Artikel der PICAXE-Serie verwendet. **Bild 3** zeigt die Transistorschaltung des Autors. Der Aufbau mit dem einfachen Schalter ist in **Bild 4** zu sehen. Der erforderliche Code für den PICAXE ist in **Listing 1** enthalten. Er kann als Textdatei direkt von der Support-Seite der Artikel-Serie auf Elektor.Labs [6] heruntergeladen werden. **Bild 5** schließlich zeigt die gesamte Schaltung mit der Drucktaste, wie sie auf einem Steckboard montiert ist. Die LED leuchtet auf, wenn Sie die Taste drücken.
Wenn Sie den Transistor verwenden, um in Ihrem Design eine Last zu schalten, müssen Sie auch sicherstellen, dass die maximale Spannung und der maximale Kollektorstrom (I_C) des Transistors nicht überschritten werden. Faktoren wie die Schaltgeschwindigkeit sind weniger relevant für langsame Schaltvorgänge.
Um den Widerstand zu berechnen, der den Strom durch die Transistor-Basis sicher begrenzt, muss sichergestellt sein, dass einerseits der maximal erlaubte Strom aus dem Controller-Pin unterschritten und andererseits der minimale Wert, der den Transistor in die Sättigung treibt, überschritten wird. Hier kommt der Verstärkungsfaktor ins Spiel: Um den minimalen Strom durch die Basis zu berechnen, dividieren Sie den Laststrom durch den Stromverstärkungsfaktor (hFE = 200). Wenn Sie sicher sein wollen, dass der Transistor wirklich voll gesättigt wird, sollten Sie den Wert mindestens verdoppeln (manchmal kann sogar eine Verfünffachung erforderlich sein!).

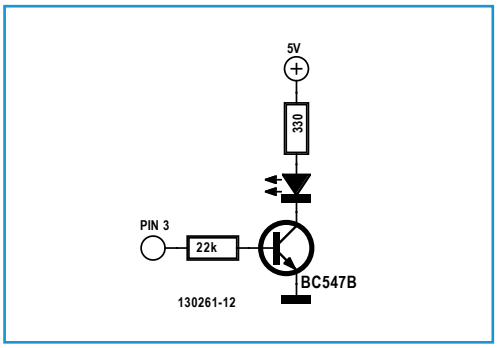


Bild 3.
Transistorschaltung.

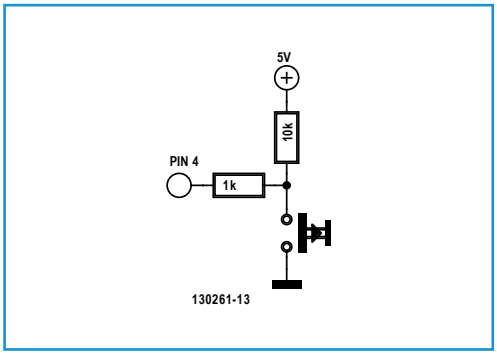


Bild 4.
Drucktasterschaltung, wie sie schon im Einführungsartikel beschrieben wurde.

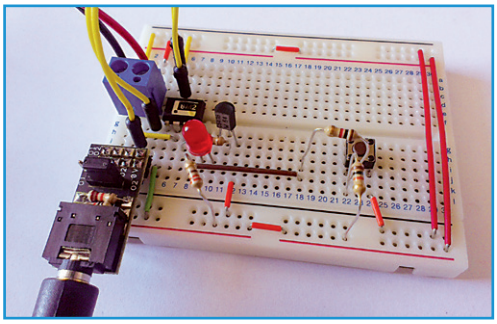


Bild 5.
Versuchsaufbau mit Transistor- und Drucktasterschaltung auf einem Steckboard.

Listing 1: Switch code

```
do                  ;repeat forever
  if pin3=0 then    ;if the button is pressed
    high 4          ;output high
  else              ;if the button is NOT pressed
    low 4           ;output low
  endif             ;closes if statement
loop
```

Für eine LED mit einem Laststrom von 10 mA sollte der Strom durch die Basis 10 mA/200 = 0,05 mA betragen, verdoppelt auf 0,1 mA. Dies ist 100 Mal weniger als der Mikrocontroller liefern kann. Berechnen Sie nun den Widerstand wie im Spannungsteiler-Beispiel: Ausgehend von 5 V fallen am Transistor (wie bei einer Diode [7]) 0,7 V ab. Damit bleiben 4,3 V für den Strombegrenzungswiderstand übrig: R = U/I = 4,3 V/0,1 mA = 43 kΩ. Zieht die LED (oder der Verbraucher) mehr als 10 mA, muss man den Widerstand kleiner wählen. Auch wenn der erlaubte Maximalstrom von 200 mA durch den Transistor fließen soll, kann der Mikrocontroller den dafür notwendigen Basisstrom von 2*200 mA/200 = 2 mA leicht verkraften. Der dafür erforderliche Widerstand wäre R = 4,3 V/2 mA = 2,2 kΩ.
Eine praktische Überlegung: Hätte ich nur 1-kΩ-Widerstände, aber keine von 2,2 kΩ, könnte ich auch zwei 1-kΩ-Typen hintereinander schalten oder – wenn ich mir keine Gedanken über den Stromverbrauch (etwa bei Batterieversorgung) machen müsste – auch nur einen 1-kΩ-Widerstand verwenden. Der Controller kann den Strom locker verkraften. In vielen Beispielschaltungen werden deshalb 1-kΩ-Widerstände sowohl zur Begrenzung des Basisstroms als auch für Low-current-Lasten direkt verwendet. Andere Transistoren können aber niedrigere Stromverstärkungsfaktoren aufweisen!

Darlington-Paare

Im Jahr 1953 stellte Sidney Darlington seine Idee vor, die Verstärkung eines Transistors zu erhöhen, indem man einen zweiten Transistor [8] hinzufügt, um sicherzustellen, dass sich die Anordnung als Schalter und nicht wie ein Verstärker verhält. Die Verstärkungsfaktoren beider Transistoren werden hierbei multipliziert (für hochverstärkende Transistoren), siehe **Bild 6**. Es ist keine Überraschung, dass mehrere PICAXE-Projektboards (wie das AXE002U, PICAXE-18M2 Starter Pack) Darlingtons als Schalter für Peripheriebausteine aufweisen, die höhere Ströme erfordern.
Bauen Sie ein Darlington-Paar nach Bild 6 auf und fügen den Schalter aus Bild 4 hinzu. Danach programmieren Sie den Controller mit dem inzwischen bekannten Code aus Listing 1. Den fertigen Versuchsaufbau zeigt **Bild 7**.
Die Auswahl von Transistoren für ein Darlington-Paar geschieht ähnlich wie bei einzelnen Transistoren, aber bedenken Sie, dass nun zwei BE-Strecken einen Spannungsabfall verursachen. Zwei Transistoren auf diese Weise umzuschalten ist zudem ein wenig langsamer als bei einem Transistor. Ob die Schaltgeschwindigkeit überhaupt ein Thema ist, hängt von der Anwendung ab. Die Berechnung des Strombegrenzungswiderstands für 2 mA geschieht wie zuvor, allerdings muss jetzt von 5 V - 0,7 V - 0,7 V = 3,6 V ausgegangen werden. So beträgt der Widerstand 3,6 V/2 mA = 1,8 kΩ, jedoch ist die Stromverstärkung viel höher (200*200 = 40.000). Der Basisstrom könnte deshalb 200 Mal niedriger sein, was einen Widerstand von 360 kΩ

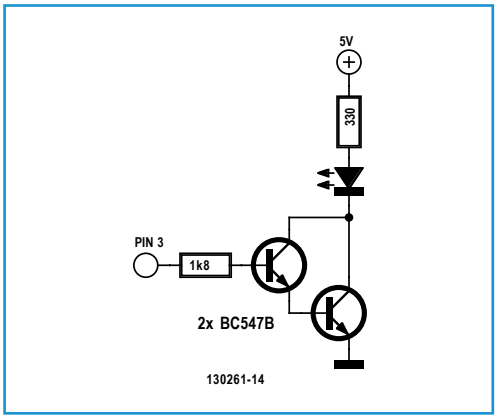


Bild 6.
Ein Darlington-Paar.

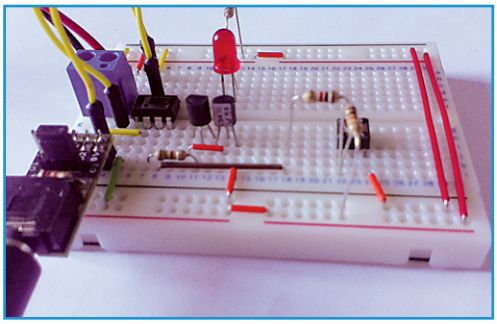


Bild 7.
Darlington-Paar auf dem Steckboard.

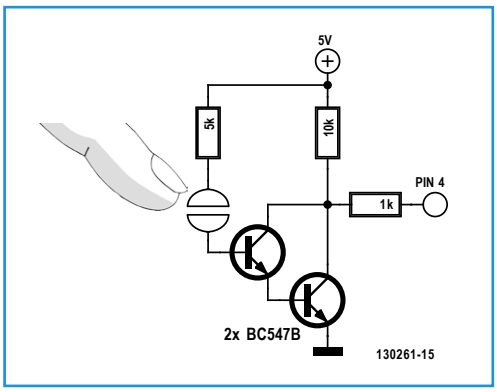


Bild 8.
Das Darlington-Paar wird als berührungsempfindlicher Schalter eingesetzt.

erfordern würde.
Solche Widerstandswerte liegen deutlich über dem höchstmöglichen Widerstand der Haut der menschlichen Fingerkuppen [9]. Deshalb lässt sich ein Darlington-Paar auch als berührungsempfindlicher Schalter wie in der Schaltung in **Bild 8** dargestellt einsetzen. Der erforderliche PICAXE-Code bleibt gleich. Vielleicht möchten Sie Strom durch die Basis und die Finger beschränken, um Schäden bei einem Kurzschluss durch unbeabsichtigtes Berühren mit einem leitenden Gegenstand zu vermeiden und um sicherzustellen, dass keine schmerzhaften Ströme >4 mA (für weitere Informationen siehe [10]) durch die Fingerspitze fließen (obwohl dies bei einer 5-V-Quelle unter normalen Bedingungen sehr unwahrscheinlich ist [11]). Dann sollten Sie den Strom (auf zum Beispiel 1 mA) mit einem Widerstand auf 5 V/1 mA = 5 kΩ begrenzen.
Bauen Sie die Schaltung mit dem Darlington-Paar als Touch-Schalter nach Bild 6 als Ausgang und einem Eingang nach Bild 8 auf. Dieses Beispiel nutzt ebenfalls den Code aus Listing 1. **Bild 9** zeigt die fertige Schaltung auf einem Steckboard (**Bild 10**) in Aktion!

Der Optokoppler

In manchen Fällen ist es von Nachteil, den Mikrocontroller aus der gleichen Stromversorgung zu betreiben wie das zu steuernde Gerät. Deshalb ist es sinnvoll, einen Optokoppler zur verwenden, um eine galvanische Trennung zu erreichen. Beide Schaltungen können eine eigene Stromversorgung besitzen, sind aber in keiner Weise „körperlich“ miteinander verbunden, sondern mit Hilfe von Licht. Ein solcher Optokoppler besteht aus einer LED und einem Fototransistor in einem Gehäuse. Wenn die LED eingeschaltet ist, kann durch den Fototransistor Strom für die zweite Schaltung fließen. Andernfalls wirkt der Optokoppler wie ein geschlossener Schalter. Einige Typen eignen sich nur für (binäre) Schaltanwendungen, andere ermöglichen die Übertragung analoger Signale.
Die Auswahl des richtigen Optokopplers kann schwierig sein, es gibt keinen einfachen Weg (dessen sich der Autor bewusst ist), um den perfekten Optokoppler für das individuelle Design aus der Unzahl der Typen, die man im Internet finden kann und die auch tatsächlich erhältlich sind, mit sehr unterschiedlichen Spezifikationen herauszusuchen. Wenn Sie jemanden fragen können, der Erfahrung in der Auswahl dieser Bauteile hat, fragen Sie ihn/sie und ersparen sich stundenlanges Surfen in langweiligen Datenblättern.
Für dieses Beispiel verwende ich den 4N25. Dieser Optokoppler dürfte wahrscheinlich für die meisten (Low-voltage-) Anwendungen funktionieren, aber konsultieren Sie das Datenblatt, um etwas über die Spannung/

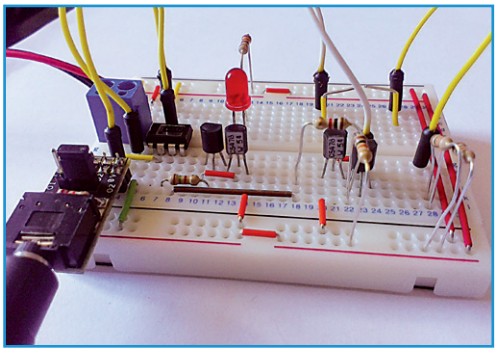


Bild 9.
Schaltung mit Touch-Schaltern auf dem Steckboard.

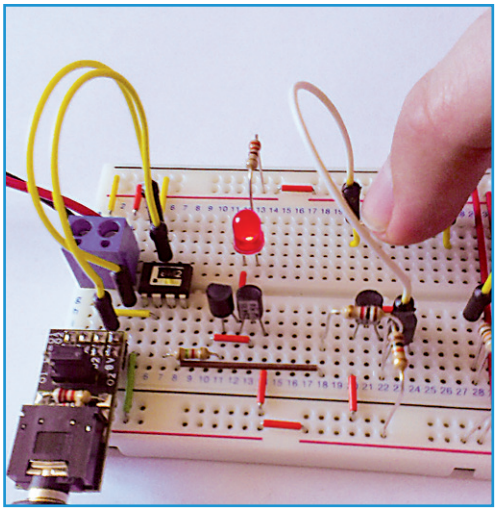


Bild 10.
Der Touch-Schalter in Aktion.

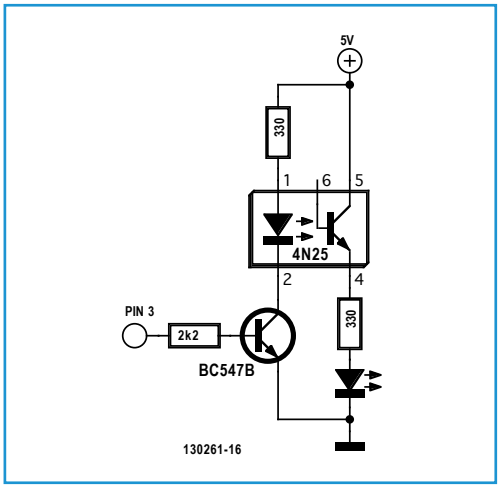


Bild 11.
Schaltung eines Optokoppler-Aufbaus.

Strom-Werte sowohl bei der LED als auch beim Fototransistor herauszufinden. Auch die Schaltgeschwindigkeit könnte für Ihre Anwendung von Bedeutung sein.
Bauen Sie die Schaltung aus **Bild 11** als Ausgang auf und verwenden Sie als Eingang den Schalter aus Bild 4 oder die berührungsempfindliche Version in Bild 8 mit dem gleichen Beispiel-Code aus Listing 1. Die fertige Schaltung auf dem Steckboard ist in **Bild 12** zu sehen.
In dem Beispiel wird ein Transistor verwendet, um den Optokoppler zu schalten, aber in den meisten Fällen kann der Optokoppler direkt (beziehungsweise über einen Widerstand zur Strombegrenzung) an den Mikrocontroller angeschlossen werden. Um den Strombegrenzungswiderstand zu berechnen, wird von einer Durchlassspannung von 1,3 V ausgegangen. Mit einem Strom von etwa 10 mA [12] ergibt sich für den Widerstand R = (5 V - 1,3 V)/0,01 A = 370 Ω, der nächstmögliche E6-Widerstand ist 330 Ω.

Relais

Wenn Sie netzbetriebene Geräte mit Ihrem PICAXE steuern möchten, muss die Netz-Schaltung von der PICAXE-Stromversorgung getrennt bleiben. Dies kann mit einem Relais geschehen, das mit Hilfe eines Elektromagneten physisch einen Schalter öffnet und schließt (zum Beispiel für eine Schreibtisch-Lampe). Legt man eine Spannung an den Elektromagneten, ändert dies den Zustand des Schalters, trennt man die Spannung wieder vom Magneten, fällt der Schalter wieder in seine ursprüngliche Stellung zurück (bei den meisten Relaistypen). Je nachdem, wie das Relais mit der Netzschaltung verbunden ist, kann die Schalterstellung standardmäßig entweder ein- oder ausgeschaltet sein. Bi-stabile Relais „erinnern“ sich an ihren Zustand, so dass kein Strom verbraucht wird, um sie im aktiven Zustand zu halten, sondern nur, um den Zustand zu wechseln. Relais sind mechanische Vorrichtungen mit einer begrenzten Anzahl von Schaltspielen, bevor sie ausfallen. Damit sind sie natürlich ungeeignet für schnell hin- und herschaltende Anwendungen. Solid-State-Relais (keine beweglichen Teile, ähnlich Optokoppler) können anstelle der herkömmlichen Relais verwendet werden, um die mechanischen Hindernisse zu überwinden, sind aber deutlich teurer.
Wenn ein Relais schließt, verursacht es eine

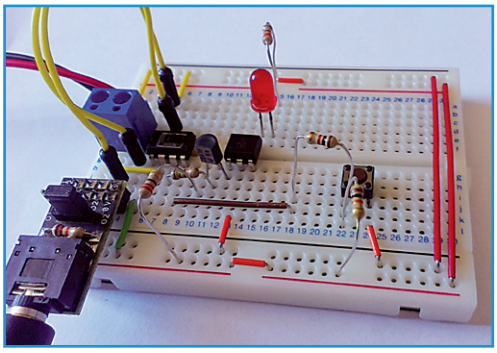


Bild 12. Optokoppler-Schaltung inklusive eines einfachen Tasters auf dem Steckboard.

Listing 2: On/Off code

```
b0 = 0              ;set state 0 (OFF)
b1 = 0              ;reset last pin state
low 4               ;reset output
main:
do                  ;repeat forever
  if pin3=0 then    ;button is pressed
    gosub btnpress  ;jump to this label
  else
    if b1=1 then    ;check button release
      b1=0          ;reset pin state variable
    endif
  endif
loop

btnpress:           ;btnpress label (for gosub)
  if b1=0 then      ;is this the first visit since
                    ;button release?
    b1=1            ;mark as visited
    gosub setstate  ;jump to label
    pause 100       ;prevent switch bounce
  endif
return              ;return to gosub statement

setstate:           ;setstate label
  if b0=0 then      ;if current state is OFF
    b0=1            ;remember new state
    high 4          ;set pin 4 high
  else              ;current state is ON
    b0=0            ;set state OFF
    low 4           ;set output pin low
  endif
return
```

Spannungsspitze, die schädlich für Elektronik sein kann. Deshalb ist eine Schutzdiode zu empfehlen, die in Sperrrichtung parallel zum Relais angeordnet ist. Diese so genannte Freilaufdiode schließt die Spannungsspitze kurz. Relaisspulen erfordern oft eine höhere Spannung, als die Stromversorgung des Mikrocontrollers liefert. Alles in allem könnte es am besten sein, ein Relais über einen Optokoppler statt nur über einen Transistor zu steuern, um die Stromversorgung des Relais (und schädliche Spannungsspitzen) vom Mikrocontroller fernzuhalten.

Alle bisherigen Beispiele haben einen Taster verwendet, um eine Ausgabe zu initiieren, aber bei einem Relais wäre ein Schalter erforderlich, um das Relais ein- und auszuschalten. Dazu müsste auch der Push-Button-Code aus Listing 1 geändert werden. Der Schaltzustand (on/off) muss deshalb gespeichert werden, nachdem die Taste losgelassen wurde. Es gibt eine Reihe von Möglichkeiten, um Daten auf dem PICAXE zu speichern, die wir aber erst in den nächsten Folgen der Artikelserie kennenlernen werden. Zunächst wollen wir die „Allzweck"-Variablen verwenden. Jeder PICAXE-Chip kennt eine Anzahl davon, mindestens 14 (genannt b0 bis b13), und jede ist 8 bit breit. Wenn Sie einen Wert mit 16 bit Breite speichern müssen, es ist möglich, zwei Allzweck-Variablen zusammen zu verwenden. In diesem Fall wäre w0 b1 und b0 zusammen, w1 dann b2 und b3 und so weiter. Beachten Sie, dass das Schreiben von w1 auch b3 und b2 überschreibt! Bei einigen neueren PICAXE-Chips ist es möglich, vier 8-bit-Variablen zu einer 32-bit-Variablen zu kombinieren. Einige Variablen werden immer zur Speicherung wichtiger Daten von vorprogrammierten PICAXE-Funktionen eingesetzt.

Im Beispiel-Code in **Listing 2** wird die Variable b0 verwendet, um entweder eine 0 (AUS-Zustand) oder eine 1 (EIN-Zustand) zu speichern. Der Ausgang wird entweder High oder Low. Beachten Sie die zusätzliche Verzögerung nach dem Ändern des Status als Reaktion auf das Drücken des Schalters. Dies ist nötig, um Schalterprellen zu unterdrücken, eine mechanische Eigenschaft der Schalter: Wenn der Taster gedrückt und wieder losgelassen wird, schaltet er für kurze Momente schnell zwischen offener und geschlossener Position hin und her. Die Variable b1 wird verwendet, um den Zustand des Eingangs-Pins

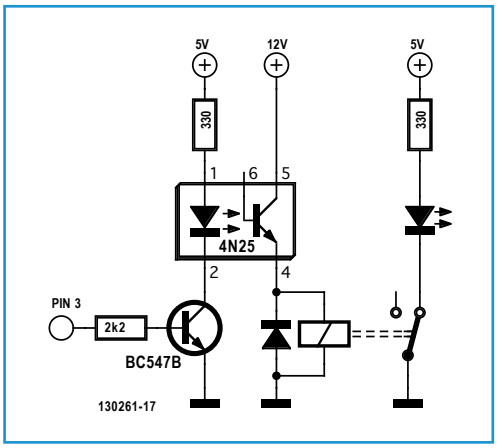


Bild 13.
Die Relaisschaltung arbeitet auch über einen Optokoppler.

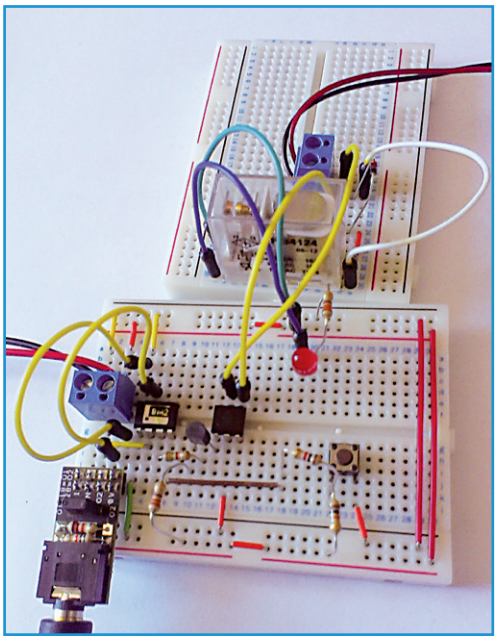


Bild 14.
Die Relaisschaltung auf einem zweiten Steckboard in Aktion.

zu speichern. Dies ermöglicht es uns, nur den Schaltzustand nach einem Loslassen/Drücken-Zyklus zu ändern, ansonsten würde die ON/OFF-Umschaltung so lange fortgesetzt, wie die Taste gedrückt wird. Der Code verwendet die `gosub`-Anweisung, um zu einem Label zu springen, in dem der Code bis zum `Return`-Befehl ausgeführt wird. Danach springt das Programm wieder in die ursprüngliche Position.

Die Relais-Beispielschaltung ist in **Bild 13** dargestellt. Mit dem Schalter aus Bild 4 sieht das auf dem Steckboard wie in **Bild 14** aus. Der Code entspricht Listing 2.

Der Optokoppler hat es nicht nötig, durch einen Transistor eingeschaltet zu werden, da der Mikrocontroller in der Lage ist, den gewünschten Strom zu liefern. In diesem Bei-

spiel wird das Relais über eine eigene Stromversorgung betrieben. Auch die Masse hat keinen Kontakt, weder zur Mikrocontroller-Schaltung noch zum Relaisschalter und damit der Schaltung hinter dem Relais. Das Beispiel verwendet +5V und GND (GND und GND3 sind in diesem Fall verbunden) der Mikrocontroller-Schaltung für die LED-Schaltung hinter dem Relais, aber abhängig vom Relais können auch netzstrombetriebene Anwendungen mit der gleichen Schaltung gesteuert werden. Wenn Sie diese Netzspannungs-Applikation aufbauen wollen, tun Sie es so sicher wie möglich (wenn Sie nicht wissen, was beim Umgang mit der Netzspannung zu beachten ist, sollten Sie es bleiben lassen). Entfernen Sie die LED- und Stromversorgungsschaltung vom Relais-Ausgang (und vor allem, überzeugen Sie sich, dass das Relais tatsächlich Netzspannung schalten kann!). Für den Aufbau können Sie ein Verlängerungskabel „opfern“. Schneiden Sie eine Ader auf und verbinden Sie diese Enden mit den Relais-Schaltkontakten, so kurz wie möglich. Zunächst sollten Sie testen, ob das Relais schaltet, ohne die Netzspannung anzuschließen. Nach dem erfolgreichen Test verbinden Sie das Kabel mit irgendeinem „Gerät“, das die Kenndaten des Relaiskontakts nicht überschreitet (zum Beispiel einer Glühbirne für einen „offensichtlichen“ Effekt) und stecken Sie den Stecker in die Steckdose. Nun sollten Sie in der Lage sein, den netzbetriebenen Verbraucher mit dem PICAXE-Schalter zu steuern. Ein Home-Automation-System ist jetzt nur ein paar Schritte entfernt.
Bei der Auswahl eines Relais ist es wichtig zu beachten, dass die tatsächlichen Werte von Spannung und Strom nicht die Grenzwerte überschreiten. Beim Schalten eines Relais mit einem Optokoppler sollte man auch immer darauf achten, dass dieser in der Lage ist, Strom und Spannung für die Relaisspule zu liefern.

Lasst uns Feierabend machen!

Bisher haben wir abgehandelt, wie PICAXE verdrahtet und programmiert wird, inklusive einer Ein- und Ausgangsschaltung. Wir wissen, was man bei der Auswahl der Bauteile für unsere eigenen Entwürfe beachten muss und wir wissen, wie man Widerstandswerte berechnet, um den Strom zu begrenzen und diese Komponenten so zu schützen. In der nächsten Folge geht es um erweiterte Ein- und Ausgänge, auch um die Ansteuerung von Servos (zum Beispiel für den Einsatz in Robotik-Schaltungen). In den folgenden Artikeln werden wir die Möglichkeiten untersuchen, integrierte Schaltungen an den PICAXE anzuschließen und so seine Fähigkeiten ins Unermessliche zu steigern. Sogar die Anbindung von PICAXE an einen PC ist möglich. Klingt gut, nicht wahr?

(130261)

Weblinks & Literatur

[1] www.picaxe.com

[2] Elektor.POST Projekt Nr. 8: „BASIC für PICs“ www.elektor-magazine.com/de/extra-downloads/post.html

[3] www.picaxe.com/Getting-Started/PICAXE-Manuals

[4] www.techsupplies.co.uk/PICAXE

[5] www.elektronik-kompendium.de/sites/grd/0201113.htm

[6] www.elektor-labs.com/picaxe

[7] www.farnell.com/datasheets/410427.pdf

[8] http://de.wikipedia.org/wiki/Darlington-Schaltung

[9] C.J. Poletto and C.L. van Doren, „A high voltage, constant current stimulator for electrocutaneous stimulation through small electrodes“, IEEE Trans biomed. Eng., Vol. 46, No. 8, S. 929-936, 1999

[10] F.A. Saunders, „Electrocutaneous displays“ in Proc. Conf. Cutaneous Commun. Syst. Devices, S. 20-26, 1973

[11] K.A. Kaczmarek, „Optimal electrotactile stimulation waveforms for human information display“, PhD thesis, Dep. Elev. Eng. Univ. Wisconsin-Madison, 1991

[12] www.vishay.com/docs/83725/4n25.pdf