

Begraaf de strijdbijl en graaf de AXE op (2)

I/O-schakelingen voor de PICAXE

In deze artikelserie behandelen we verschillende aspecten van het ontwerpen van elektronica-projecten op basis van PICAXE-microcontrollers. Dit tweede artikel gaat over schakelingen om een PICAXE-chip te voorzien van inputs en outputs met standaard elektronica-componenten.



Het eerste artikel in deze serie was een inleiding in het PICAXE-systeem [1], met voorbeelden voor het bouwen van een programmer voor de PICAXE. We hebben toen ook laten zien hoe we de PICAXE-chip kunnen programmeren om een LED te besturen met een drukknop. Dit artikel gaat over schakelingen om een PICAXE-chip te voorzien van inputs en outputs met standaard elektronica-componenten. We willen in dit artikel ook laten zien, welke factoren en berekeningen een rol spelen bij de keuze van componenten voor eigen interface-schakelingen. In de volgende artikelen zullen we ingaan op meer geavanceerde, chip-specifieke interface-mogelijkheden voor de PICAXE. De mogelijkheden van een project gebaseerd op een PICAXE-chip kunnen worden uitgebreid door het toevoegen van speciale IC's en periferie, zoals extra geheugen, keyboards, LC-displays of zelfs een seriële verbinding met een PC.

**Alle kennis is beschikbaar...
Maar waar?**

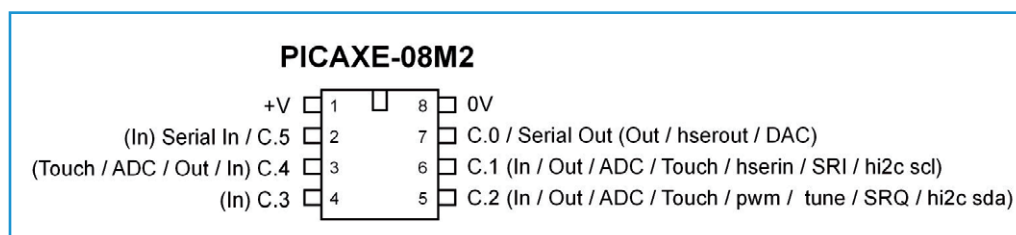
Op het Internet is enorm veel informatie beschikbaar voor iedereen die zelf microcontroller-gebaseerde elektronica wil gaan ontwerpen. Helaas kan het heel moeilijk zijn om

voldoende informatie te vinden over de onderliggende theorie. Die staat vaak niet op dezelfde plek als de specifieke voorbeelden van interface-schakelingen: het is gemakkelijk te vinden welke componenten gebruikt worden en hoe die moeten worden aangesloten, maar er staat niet bij, waarom nou precies die componenten zijn gekozen. Bij de volgende voorbeelden van interfaces zeggen we ook iets over de theorie, zodat duidelijk wordt hoe de juiste componenten voor eigen ontwerpen gekozen kunnen worden.

Vorbereiding: opbouw van een experimenteersysteem

Alle experimenten in dit artikel zijn uitgevoerd met een PICAXE 08M2 (zie **figuur 1** voor de penconfiguratie). De schakeling wordt gevoed met 5 V uit een ATX-voeding. De voorbeeldschakelingen worden opgebouwd op breadboards, dus zonder solderen. De PICAXE-chips worden geprogrammeerd via een USB-naar-serieel-kabel met de methode die is beschreven in het vorige artikel [2]. Hierbij wordt LinAXEpad-software versie 1.5.0 voor (Arch)-Linux gebruikt. Om verwarring te

Wouter Spruit
(Nederland)



Figuur 1.
PICAXE 08M2
penconfiguratie.

vermijden, verwijzen alle pennummers in de schema's naar de fysieke pennummers op de chip (bij alle voorbeelden: pen 3 is een output, pen 4 is een input), de pennummers in de listings verwijzen naar interne pennamen volgens **figuur 1**. Zie het vorige artikel [2], of de PICAXE-handleiding op de website [3] voor voorbeelden van het programmeren van een PICAXE-chip. PICAXE-chips en randapparatuur zijn verkrijgbaar via de webwinkel van Revolution Education [4].

Bescherm de componenten

Componenten hebben maximaal toelaatbare spanningen en stromen. Bij overschrijding van die maximumwaarden worden ze waarschijnlijk onherstelbaar beschadigd. Het verband tussen stroom, spanning en weerstand ligt vast in de Wet van Ohm: Spanning (V) = Stroom (I) * Weerstand (R). Dat betekent dat de stroom door een component rechtstreeks bepaald wordt door het spanningsverschil tussen twee punten en de weerstand tussen die punten. Bij een vaste gelijkspanningsbron, zoals de 5 V-voeding die we hier gebruiken, kan de stroom dus beperkt worden door een weerstand in serie te schakelen. Als we weerstanden in serie schakelen, neemt de totale weerstand toe ($R = R_1 + R_2 + R_3 + \dots + R_n$). Als we weerstanden parallel schakelen, neemt de totale weerstand af ($R = 1/R_1 + 1/R_2 + 1/R_3 + \dots + 1/R_n$). Als we componenten in serie schakelen, loopt door allemaal dezelfde stroom, hoewel ze een verschillende weerstand kunnen hebben. Uit de Wet van Ohm volgt dat de spanning evenredig met de weerstand verdeeld wordt over de componenten. Tot zo ver zal iedereen met interesse voor elektronica wel op de hoogte zijn, maar er is op het Internet nog veel meer informatie te vinden over de Wet van Ohm, bijvoorbeeld op deze website [5]. Het is nooit te laat om er iets bij te leren!

Het volgende voorbeeld maakt het berekenen van weerstanden in echte schakelingen veel gemakkelijker. Kijk eens naar de spanningsdeler in **figuur 2**. De weerstanden R1 en R2 zijn in serie geschakeld, dus de stroom door beide weerstanden is gelijk. Maar er staat over R1 een andere spanning dan over R2. Dat ligt aan de verschillende weerstandswaarden. Omdat de Wet van Ohm gaat over spanningsverschillen, geldt dat verband tussen spanning,

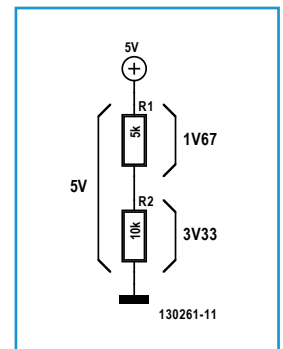
stroom en weerstand ook voor elk gedeelte van een schakeling. Zoals we zien in **figuur 2**, is de spanning over R1+R2 gelijk aan +5 V. De spanning over R1 is $+5 V * (R_1 / (R_1 + R_2)) = 5 * (5 k / 15 k) = 1,67 V$, dat volgt uit $V = R * I$. Het is logisch dat de rest van de spanning valt over R2: $5 - 1,67 = 3,33 V = 5 * (R_2 / (R_1 + R_2))$. Vergeet niet, dat dit alleen van toepassing is voor ohmse componenten, maar niet hoeft te gelden voor andere soorten onderdelen, zoals bijvoorbeeld diodes (inclusief LED's en optocoupler inputs) en transistoren. Gebruik de spanningsval over een LED om een weerstand te berekenen voor het begrenzen van de stroom. Probeer niet de weerstand van een LED te meten met een multimeter! Voorbeeld: om de stroom door een rode LED met een spanningsval van 1,8 V te beperken tot 10 mA, als hij wordt gevoed uit een spanningsbron van +5 V, berekenen we de weerstandswaarde als volgt: spanning over de weerstand = $5 V - 1,8 V$ (spanningsval over de LED) = 3,2 V. Nu gebruiken we $V/I=R$ om de weerstandswaarde te vinden: $3,2 V / 0,010 A = 320 \Omega$. De dichtstbijzijnde gangbare weerstandswaarde (in de E6-reeks) is 330 Ω .

We kunnen het principe van de spanningsdeler goed gebruiken in combinatie met de ingangen van een microcontroller. Stel dat we R1 of R2 in de schakeling van **figuur 2** vervangen door één of andere variabele weerstand, bijvoorbeeld om een temperatuur of lichtsterkte te meten. We kunnen een ingangspan van een microcontroller aansluiten op het knooppunt tussen R1 en R2 en zo met een A/D-converter een spanning meten die evenredig is met het meetsignaal.

Omdat er een grens is aan de stroom die een chip kan opnemen (afvoeren naar massa) of leveren (vanuit de voedingsspanning), is het in elke (interface-)schakeling belangrijk de stroom die van en naar de chip loopt te beperken. Maar in veel gevallen heeft de randapparatuur veel meer stroom nodig om te kunnen werken, dan de chip kan leveren.

De transistor

Een bespreking van de details van alle bestaande transistortypes valt buiten het kader van dit artikel. We laten alleen zien, hoe we met de bekende NPN-transistor van



Figuur 2. Voorbeeld van een spanningsdeler.

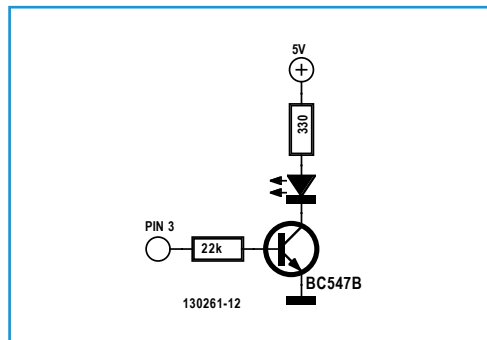
het type BC547B (of equivalent) het probleem van de beperkte stroom op de microcontroller-pennen kunnen overwinnen. We laten de transistor de stroom door de periferie schakelen.

Een NPN-transistor regelt de stroom van zijn collector- naar zijn emitter-pen afhankelijk van de spanning op zijn basis. Een kleine stroom door de basis leidt tot een grotere stroom door de collector- en emitterpen, dus de transistor werkt als een versterker, totdat hij in verzadiging raakt; dan neemt de collectorstroom niet verder meer toe. Bij een PNP-transistor lopen de stromen andersom, maar daar gaan we nu niet verder op in.

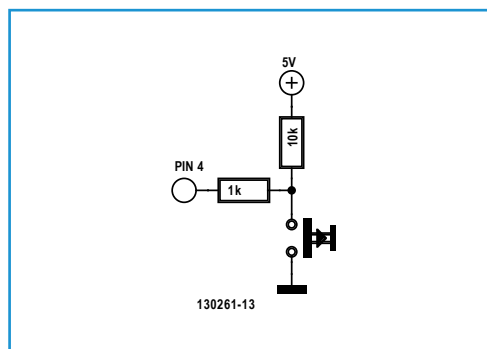
Gewapend met deze kennis gaan we een schakeling bouwen om te testen of de transistor een LED kan schakelen: de stroom uit een outputpen van de PICAXE bestuurt een transistor en die stuurt de LED aan. De outputpen van de PICAXE hoeft alleen maar de stroom te leveren om de transistor te laten schakelen; de stroom voor de LED komt in dit geval rechtstreeks van de voeding.

Dit voorbeeld werkt met dezelfde drukknop-schakeling die we in het inleidende artikel van deze serie hebben gebruikt. In **figuur 3** zien we de transistorschakeling van de auteur. Ook het circuit met de drukknop van **figuur 4** moet worden toegevoegd. De code voor de PICAXE-chip staat in **Listing 1**. De code is ook als tekstbestand te downloaden van de pagina van deze serie bij Elektor.LABS [6]. In **figuur 5** zien we de complete schakeling, met drukknop, opgebouwd op een breadboard. De LED licht op als er op de knop wordt gedrukt.

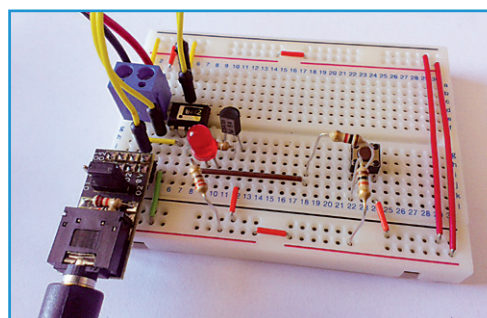
Let bij het gebruik van een transistor als schakelaar in eigen ontwerpen altijd op de maximale spanning (voor de verschillende aansluitingen) en maximale collectorstroom (I_c) van de transistor. Factoren als de schakelsnelheid en versterking (h_{FE}) zijn voor "langzame" aan/uit-toepassingen zoals deze, minder belangrijk dan voor meer veeleisende schakelingen. Bij het berekenen van de waarde van de stroombegrenzingsweerstand voor de basisstroom, moeten we zorgen dat de stroom niet groter wordt dan de maximaal toegestane waarde, maar wel groot genoeg om de transistor in verzadiging te sturen: hier gaat de versterking een rol spelen. Bereken de minimale basisstroom



Figuur 3. Transistorschakeling



Figuur 4. Drukknop-schakeling zoals besproken in het inleidende artikel.



Figuur 5. Experiment met de transistorschakeling en een drukknop op een breadboard.

door de belastingsstroom te delen door de versterkingsfactor ($h_{FE} = 200$). Voor de zekerheid is het goed om die op zijn minst te verdubbelen. Soms is zelfs het vijfvoudige nodig om zeker te stellen dat de transistor verzadigd wordt.

Als de stroom door de LED 10mA is, moet de stroom door de basis $10/200 = 0,05$ mA

Listing 1: Code voor de schakelaar

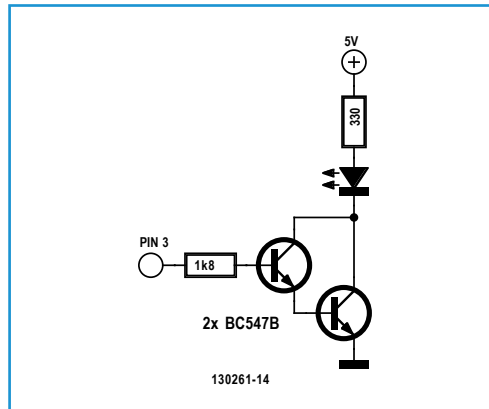
```
do                                ;repeat forever
  if pin3=0 then                  ;if the button is pressed
    high 4                        ;output high
  else                             ;if the button is NOT pressed
    low 4                          ;output low
  endif                    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

zijn. Als we dat verdubbelen, komen we uit op 0,1 mA (dus hoeft de microcontroller 100 keer minder stroom te leveren). Bereken nu de weerstandswaarde op dezelfde manier als in het voorbeeld van de spanningsdeler: We beginnen met 5 V, de transistor neemt 0,7 V, net als een diode [7], dus er staat 4,3 V over de stroombegrenzingsweerstand: $R = V/I$ dus $R = 4,3 \text{ V} / 0,1 \text{ mA} = 43 \text{ k}$. Maar voor een belasting die meer dan 10 mA trekt moet de weerstand kleiner zijn. Voor het gemak kiezen we meteen een kleinere weerstand, zodat de maximale stroom door de transistor 200 mA kan zijn. De microcontroller kan gemakkelijk de $2 \cdot 200 / 200 = 2 \text{ mA}$ verwerken die daarvoor nodig is. Voor deze weerstand: $R = 4,3 / 2 \text{ mA} = 2,2 \text{ k}$. Een praktische opmerking: Stel dat we toevallig geen weerstand van 2,2 k in huis hebben, maar wel een heleboel van 1 k. Dan kunnen we gewoon twee weerstanden van 1 k in serie zetten. En als het energieverbruik niet belangrijk is (omdat de schakeling niet op batterijvoeding hoeft te werken), kunnen we ook gewoon één weerstand van 1 k kiezen, als de microcontroller de stroom aankan. Misschien is dat de reden dat we in veel voorbeeldschakelingen een weerstand van 1 k zien om de basisstroom te begrenzen, ook als de belasting niet veel stroom nodig heeft. Er zijn ook veel bekende transistoren die een lagere versterkingsfactor hebben!

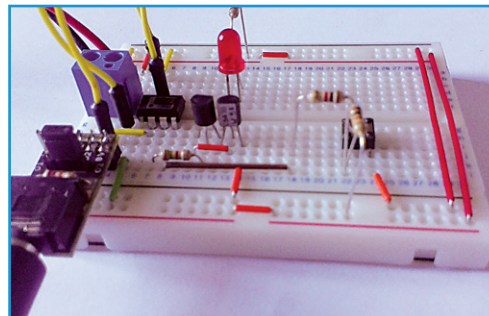
Het Darlington-paar

In 1953 kwam Sidney Darlington met het idee de versterking op te voeren, door een tweede transistor toe te voegen [8]. Zo weten we zeker dat de transistor zich gedraagt als een schakelaar in plaats van een versterker, omdat de versterkingsfactoren van de transistoren worden vermenigvuldigd (voor transistoren met een grote versterking), zie **figuur 6**. Het is dus niet zo vreemd, dat veel PICAXE-kaarten (zoals het AXE002U PICAXE-18M2 Starter Pack) Darlington-paren gebruiken als schakelaar voor het aansturen van periferie die veel stroom nodig heeft.

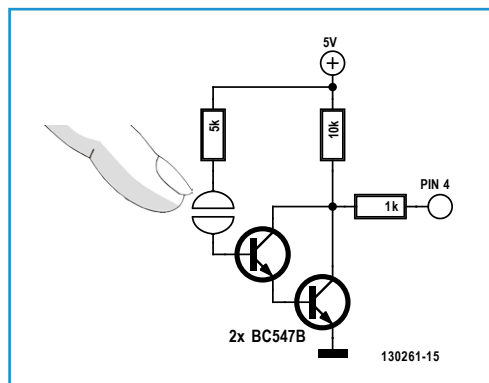
Bouw de schakeling met het Darlington-paar op volgens van figuur 6 en voeg de schakelaar van figuur 4 toe. Programmeer de schakeling met de bekende code uit listing 1. De complete opbouw van de experimentele schakeling is te zien in **figuur 7**.



Figuur 6. Schema van een Darlington-paar.



Figuur 7. Het Darlington-paar op een breadboard



Figuur 8. Een Darlington-paar gebruikt als aanraakschakelaar.

Het kiezen van transistoren voor een Darlington-paar lijkt op het kiezen van losse transistoren, maar let er op dat de spanning over de beide BE-overgangen bij elkaar opgeteld wordt. De twee transistoren samen schakelen wel iets langzamer dan één transistor (of de schakelsnelheid er toe doet is afhankelijk van de toepassing). Als we nu weer de stroombegrenzingsweerstand voor 2mA berekenen, moeten we uit gaan van een spanning van $5 - 0,7 - 0,7 = 3,6 \text{ V}$, zodat we uitkomen op $3,6 \text{ V} / 2 \text{ mA} = 1,8 \text{ k}$, maar de stroomversterking is nu veel hoger ($200 \cdot 200 = 40.000$), dus de basisstroom van 2 mA mag nu 200 keer kleiner zijn. Dus komen we uit op $1,8 \text{ k} \cdot 200 = 360 \text{ k}$.

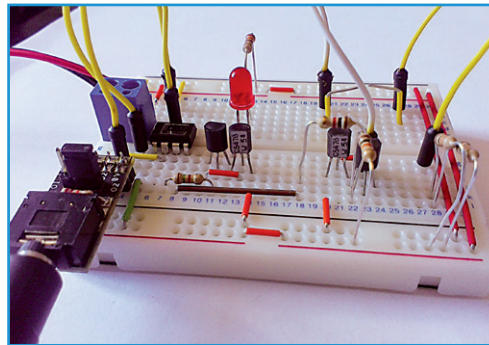
Omdat die weerstandswaarde veel groter is dan de weerstand van de huid van een menselijke vingertop [9], kunnen we een Darlington-paar ook gebruiken voor een aanraak-schakelaar, zoals in **figuur 8**. De PICAXE-code blijft in dit geval hetzelfde. Het is mogelijk de stroom door de basis en de vingercontacten te begrenzen om te voorkomen dat er bij kortsluiting door een geleidend voorwerp schade ontstaat of dat er pijnlijke stromen (>4 mA, voor meer informatie zie [10]) door de vingertop vloeien, hoewel het heel onwaarschijnlijk is dat dat ooit gebeurt bij een voeding van 5 V [11]. Voor de zekerheid moet de weerstand de stroom door de vinger beperken tot 1 mA: $5\text{ V}/1\text{ mA} = 5\text{ k}$.

Bouw de Darlington-aanraakschakelaar volgens figuur 6 als output en de schakeling in figuur 8 als input. Dit voorbeeld werkt weer met dezelfde code uit listing 1. In **figuur 9** zien we de complete schakeling op een breadboard en in **figuur 10** zien we hem in actie.

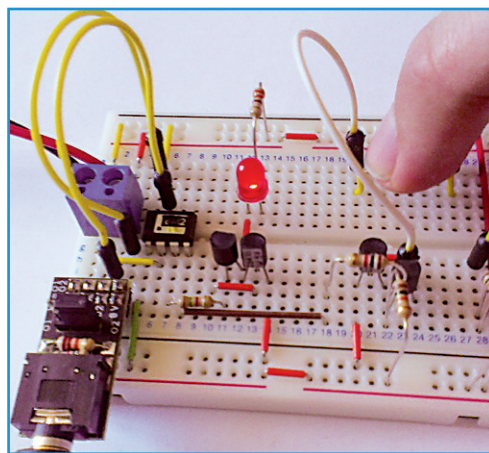
De optocoupler

Soms het is het niet mogelijk de microcontrollerschakeling van stroom te voorzien uit dezelfde voeding als het te besturen apparaat. Met een optocoupler kunnen we iets schakelen via een optische isolatie: beide schakelingen kunnen hun eigen voeding hebben, ze zijn niet elektrisch met elkaar verbonden. De koppeling wordt met licht tot stand gebracht. Een optocoupler bestaat uit een LED en een fototransistor in één behuizing. Als de LED brandt, laat de fototransistor stroom door. Als de LED uit is, blokkeert de fototransistor de stroom. Sommige optocouplers zijn alleen geschikt als binaire schakelaar, andere kunnen analoge signalen doorgeven. Het kiezen van een optocoupler kan lastig zijn, want er is geen gemakkelijke manier (voor zover de auteur weet) om de perfecte optocoupler voor een bepaalde toepassing te vinden in het brede aanbod van types (met heel verschillende specificaties) dat via Internet en in winkels verkrijgbaar is. Als het even kan, vraag dan advies van iemand met ervaring in het kiezen van een optocoupler voor een specifieke toepassing, anders moeten er heel veel data sheets doorgespiet worden.

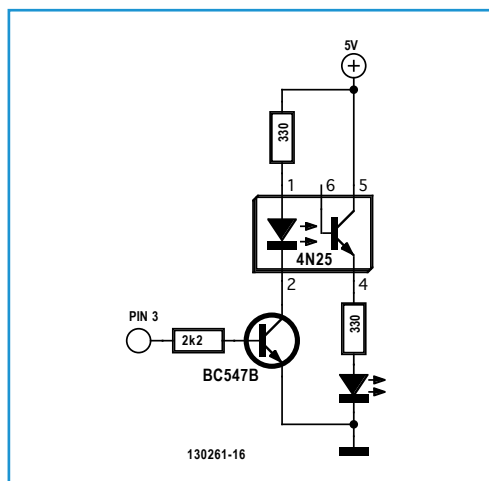
In dit voorbeeld gebruiken we de 4N25. Deze optocoupler is waarschijnlijk geschikt voor de



Figuur 9. Een aanraakschakelaar op een breadboard.



Figuur 10. De aanraakschakelaar in actie.



Figuur 11. Schema voor het toepassen van een optocoupler.

meeste (laagspannings-) toepassingen, maar controleer het data sheet op de maximale spanning en stroom van zowel de LED als de fototransistor en op de schakelsnelheid, voor zover die relevant is voor de toepassing.

Bouw de voorbeeldschakeling van **figuur 11** voor de output pen, en kies voor de input tussen de simpele schakelaar uit figuur 4 of de aanraakschakelaar uit figuur 8. Gebruik ook nu weer dezelfde voorbeeldcode uit listing 1.

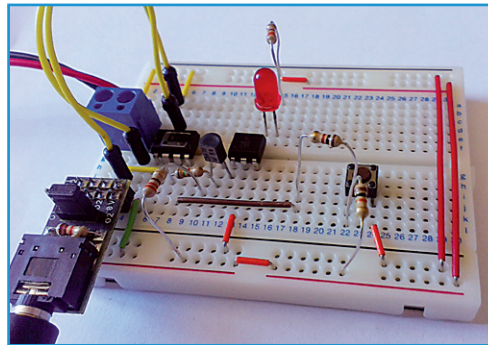
De complete opbouw van de schakeling is te zien in **figuur 12**.

In dit voorbeeld wordt een transistor gebruikt om de optocoupler te schakelen, maar meestal een kan optocoupler rechtstreeks, via een stroombegrenzingsweerstand, worden verbonden met een microcontrollerpin. Om de stroombegrenzingsweerstand te berekenen, zijn we uitgegaan van een geleidingsspanning van 1,3 V. Om uit te komen op een stroom van ongeveer 10 mA [12] berekenen we R als volgt: $R = (5\text{ V} - 1,3\text{ V}) / 0,01\text{ A} = 370\ \Omega$, afgerond 330 Ω .

Relais

Als we apparatuur die uit het lichtnet wordt gevoed willen schakelen met de PICAXE, moet het netspanningsgedeelte geïsoleerd blijven van de voeding van de PICAXE. Relais bevatten een schakelaar (bijvoorbeeld voor het schakelen van een bureaulamp) die wordt bestuurd door een elektromagneet: zet een spanning op de elektromagneet om de toestand van de schakelaar te veranderen. Haal de spanning weer weg om terug te gaan naar de begintoestand. De beginpositie van de schakelaar kan aan of uit zijn, afhankelijk van hoe het relais is verbonden met de (netspannings-) schakeling. Bistabiele relais "onthouden" hun toestand, dus er wordt geen vermogen verbruikt om de toestand vast te houden, alleen bij het schakelen. Relais zijn mechanisch. Ze kunnen tijdens hun levensduur een eindig aantal keren schakelen. Daardoor zijn ze minder geschikt voor toepassingen waar heel snel en heel veel moet worden geschakeld. In zulke gevallen kan een solid-state-relais (zonder bewegende delen, net als bij optocouplers) worden gebruikt in plaats van een traditioneel relais. Dat lost het mechanische probleem op, maar zulke relais zijn een stuk duurder.

Als een relaispoel wordt geschakeld, ontstaat er een spanningspiek die schadelijk is voor elektronica, daarom wordt er altijd een beschermingsdiode anti-parallel met de relaispoel geplaatst. Deze vrijloop diode moet onder normale omstandigheden voorkomen dat er een stroom loopt. (Controleer de maximale stroom en spanning waarvoor de diode geschikt is). Relais hebben vaak een hogere voedingsspanning nodig dan microcontrollers.



Figuur 12. Optocouplerschakeling op een breadboard met een simpele drukknopschakeling.

Al met al is het misschien wel het beste om een relais aan te sturen via een optocoupler in plaats van alleen een transistor. Dat houdt

Listing 2: Aan/uit-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

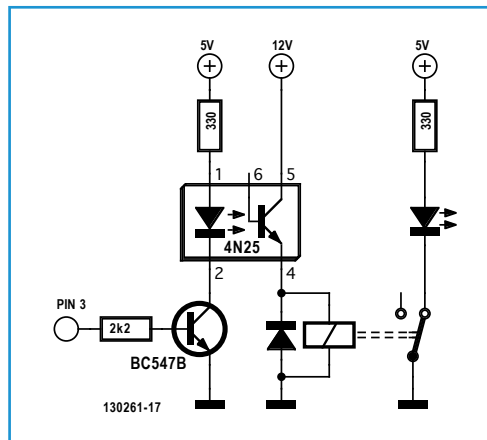
btnpress:       ;btnpress label (for gosub)
  if b1=0 then  ;is this the first visit since
                ;button release?
    b1=1        ;mark as visited
    gosub setstate ;jump to label
    pause 100   ;prevent switch bounce
  endif
return          ;return to gosub statement

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
    
```

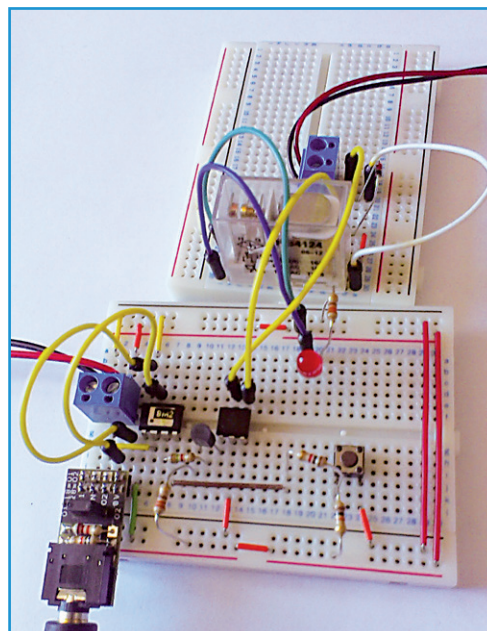
de relaisvoeding (en eventuele spanningspieken) weg van de microcontroller.

In alle vorige voorbeelden hebben we een drukknop gebruikt die alleen een uitgangssignaal geeft als op de knop wordt gedrukt, maar voor het relaisvoorbeeld kiezen we voor een omschakelaar: indrukken van de knop doet de stand wisselen tussen aan en uit. Om dat te realiseren, moeten we de code van listing 1 zó aanpassen dat de toestand (aan/uit) wordt bewaard als de knop wordt losgelaten. Er zijn verschillende manieren om data op te slaan op de PICAXE-chip. We gaan daar in de volgende artikelen op in. Voorlopig maken we gebruik van ingebouwde variabelen van de PICAXE. Elke PICAXE-chip heeft een aantal van die variabelen, tenminste 14 stuks (ze heten b0 t/m b13), en ze kunnen allemaal 8 bits opslaan. Als we een waarde in 16 bits willen opslaan, kunnen we twee variabelen combineren. Dan vormen b1 en b0 samen w0, b2 en b3 vormen w1, enzovoort. Als we dus iets naar w1 schrijven, overschrijven we b2 en b3! In sommige nieuwere PICAXE-chips kunnen we vier 8-bits variabelen combineren tot een 32-bits variabele. Sommige variablenummers worden altijd gebruikt voor het opslaan van belangrijke data die te maken heeft met een specifieke voorgeprogrammeerde PICAXE-functionaliteit.

De voorbeeldcode in **listing 2** laat zien hoe we de variabele b0 kunnen gebruiken voor het opslaan van een 0 (UIT toestand) of een 1 (AAN toestand) en hoe we die toestand kunnen omschakelen met een drukknop. Het uitgangssignaal krijgt daarbij dezelfde waarde als de variabele. Let op de extra vertraging na het veranderen van de toestand als reactie op het drukken op de knop: dit is om het zogenaamde denderen van het contact te onderwerpen. Dit denderen is een mechanische eigenschap van schakelaars: als ze aan- of uitschakelen wisselen ze korte tijd heel snel heen-en-weer tussen open en gesloten. Variabele b1 wordt gebruikt om de toestand van de input-pen op te slaan: Daardoor kunnen we de toestand alleen laten veranderen na een complete cyclus van indrukken en loslaten. Anders zou het AAN-/UIT-omschakelen doorgaan zolang de knop werd ingedrukt. Voor de leesbaarheid maakt de code gebruik van de "gosub"-instructie. "gosub" springt naar



Figuur 13. Schema van de relaischakeling. Ook het relais wordt aangestuurd via een optocoupler.



Figuur 14. De relaischakeling in actie, opgebouwd op een tweede breadboard.

een label, voert de code vanaf dat punt uit tot aan een "return" statement, waarna de uitvoering van het programma doorgaat vanaf de plek waar vandaan gesprongen was.

De schakeling voor dit voorbeeld is gebouwd met de relaischakeling in **figuur 13** en de schakelaar van figuur 4, maar dit keer moet de PICAXE worden geprogrammeerd met de code uit listing 2. De schakeling, opgebouwd op een breadboard, is te zien in **figuur 14**.

Ook nu hoeft de optocoupler niet aangestuurd te worden door een transistor, als de microcontroller voldoende stroom kan leveren. In dit voorbeeld heeft het relais zijn eigen voeding. De massa is niet verbonden met de microcontrollerschakeling en met het uit-

gangscircuit. In het voorbeeld gebruiken we +5V en GND (dus GND en GND3 worden in dit geval met elkaar verbonden) van de micro-controllerschakeling voor de LED-schakeling na het relais. Maar, afhankelijk van het relais, zouden we met deze schakeling ook netgevoede apparaten kunnen schakelen.

Let daarbij vooral op de veiligheid! (Wie niet weet wat dat betekent en waar op gelet moet worden, kan het beter niet proberen.) Koppel de LED-belasting en de voeding af van het relais (en boven alles: controleer of het relais inderdaad geschikt is om netspanning te schakelen!). Om een schakelaar te maken, kunnen we een verlengsnoer "offeren". Let op dat het niet aangesloten is! Snijd het snoer door en sluit het zó aan op de relaiscontacten, dat de verbinding hersteld wordt, als de relaiscontacten sluiten. Test dan eerst of de PICAXE-schakeling nog werkt zonder het snoer aan te sluiten. Verbind dan het snoer aan de ene kant met een verbruiker die niet te veel stroom verbruikt (een lamp bijvoorbeeld) en steek de stekker aan het andere eind in het stopcontact. Nu moet de lamp te schakelen zijn met de drukknop van de PICAXE-schakeling! We zijn nu nog maar een paar stappen verwijderd van een compleet domotica-systeem...

Let bij de keuze van een relais, op de spanning en stroom die nodig is om het aan te sturen en op de maximale spanning en stroom die het kan schakelen. Als het relais wordt aangestuurd via een optocoupler, let er dan ook op, of de optocoupler in staat is de stroom en spanning voor de relaisspoel te schakelen!

Genoeg voor vandaag

We hebben nu gezien hoe we de PICAXE-chip kunnen aansluiten en programmeren en we hebben verschillende voorbeelden van in- en uitgangscircuits uitgeprobeerd. We weten waar we op moeten letten als we componenten kiezen voor eigen ontwerpen en we weten hoe we de weerstandswaarden moeten berekenen om de stroom door die componenten te beperken. In het volgende artikel zullen we ingaan op meer geavanceerde inputs en outputs, onder meer op servo's, bijvoorbeeld voor het bouwen van robots. In latere artikelen gaan we kijken naar het koppelen van andere geïntegreerde schakelingen aan een PICAXE-project, wat geweldige nieuwe mogelijkheden biedt, zelfs besturing van de PICAXE vanuit een PC. Er staat ons dus nog veel leuks te wachten!

(130261)

Internet-links en literatuur

- [1] www.picaxe.com
- [2] Elektor.POST Project No. 8: "Begraaf de strijdbijl en graaf de AXE op" www.elektor-magazine.com/extra/post
- [3] www.picaxe.com/Getting-Started/PICAXE-Manuals
- [4] www.techsupplies.co.uk/PICAXE
- [5] <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/ohmlaw.html>
- [6] www.elektor-labs.com/picaxe
- [7] www.farnell.com/datasheets/410427.pdf
- [8] http://en.wikipedia.org/wiki/Darlington_transistor
- [9] C.J. Poletto en C.L. van Doren, "A high voltage, constant current stimulator for electrocutaneous stimulation through small electrodes", IEEE Trans biomed. Eng., vol 46, no. 8, pp. 929-936, 1999
- [10] F.A. Saunders, "Electrocutaneous displays" in Proc. Conf. Cutaneous Commun. Syst. Devices, pp 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