

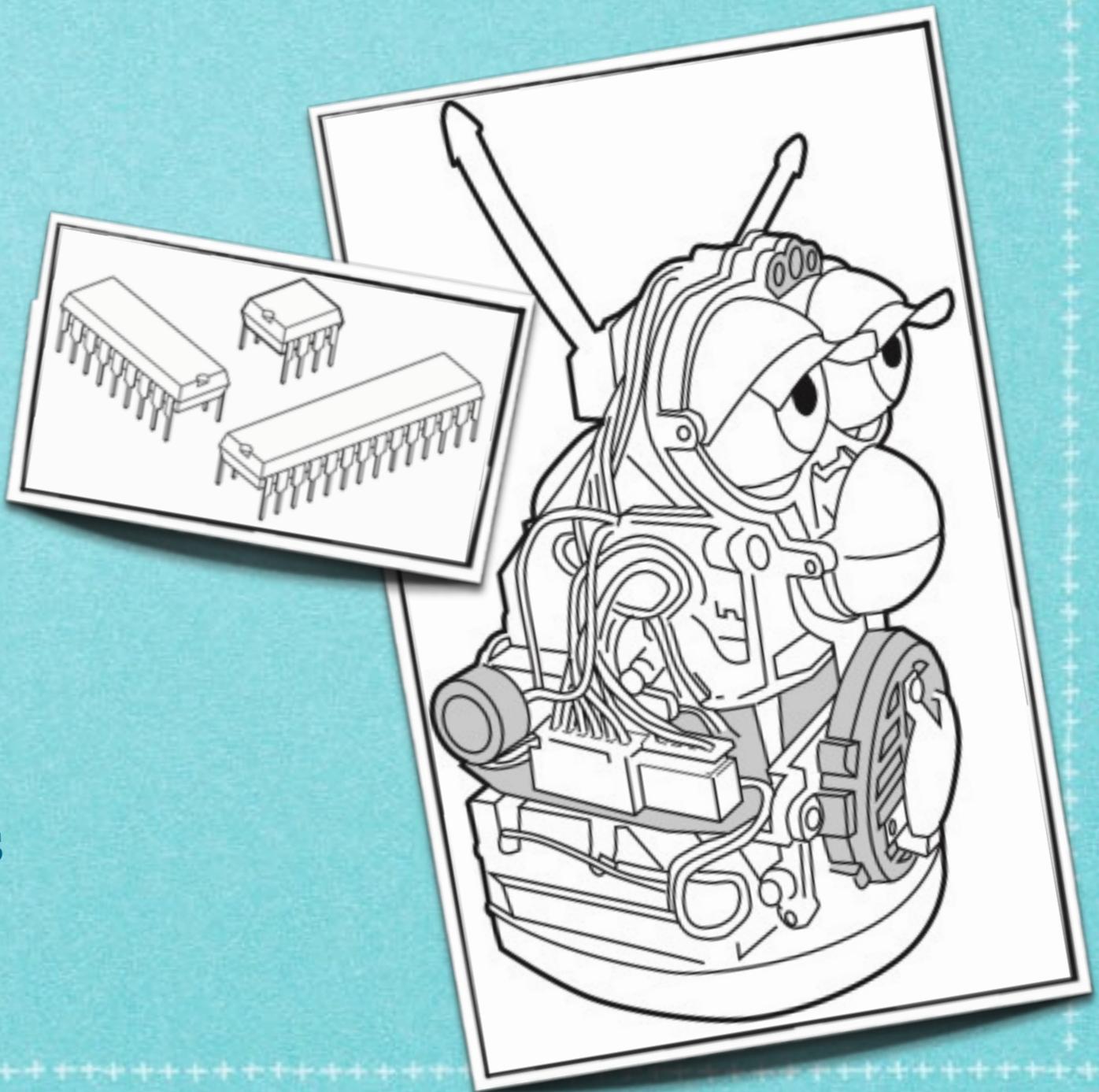
- Entdeckungen -
Elektronische Kuscheltiere

Workshop Juni 2010
Olav Schettler <olav@schettler.net>

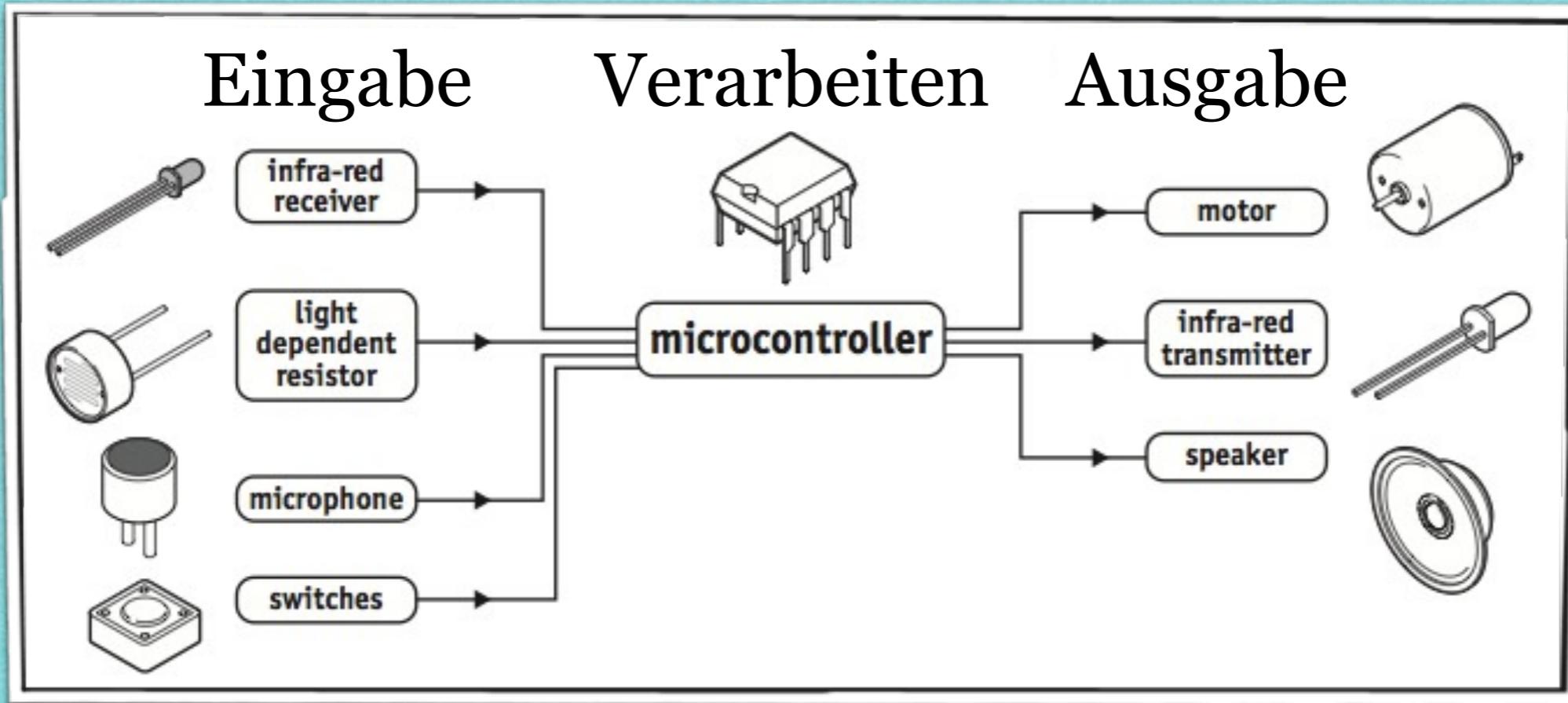


Was sind Microcontroller?

- ▶ Computer auf einem Chip
- ▶ Elektronengehirn
- ▶ Speicher, Rechenkern, Ein-/Ausgabe-Pins
- ▶ Programmierbar
- ▶ Beispiel Spielzeug „Furby“:
Berühren, Licht
=> Bewegung, Töne
- ▶ Schalter, Lichtfühler,
Mikrofon ... Motoren, LEDs



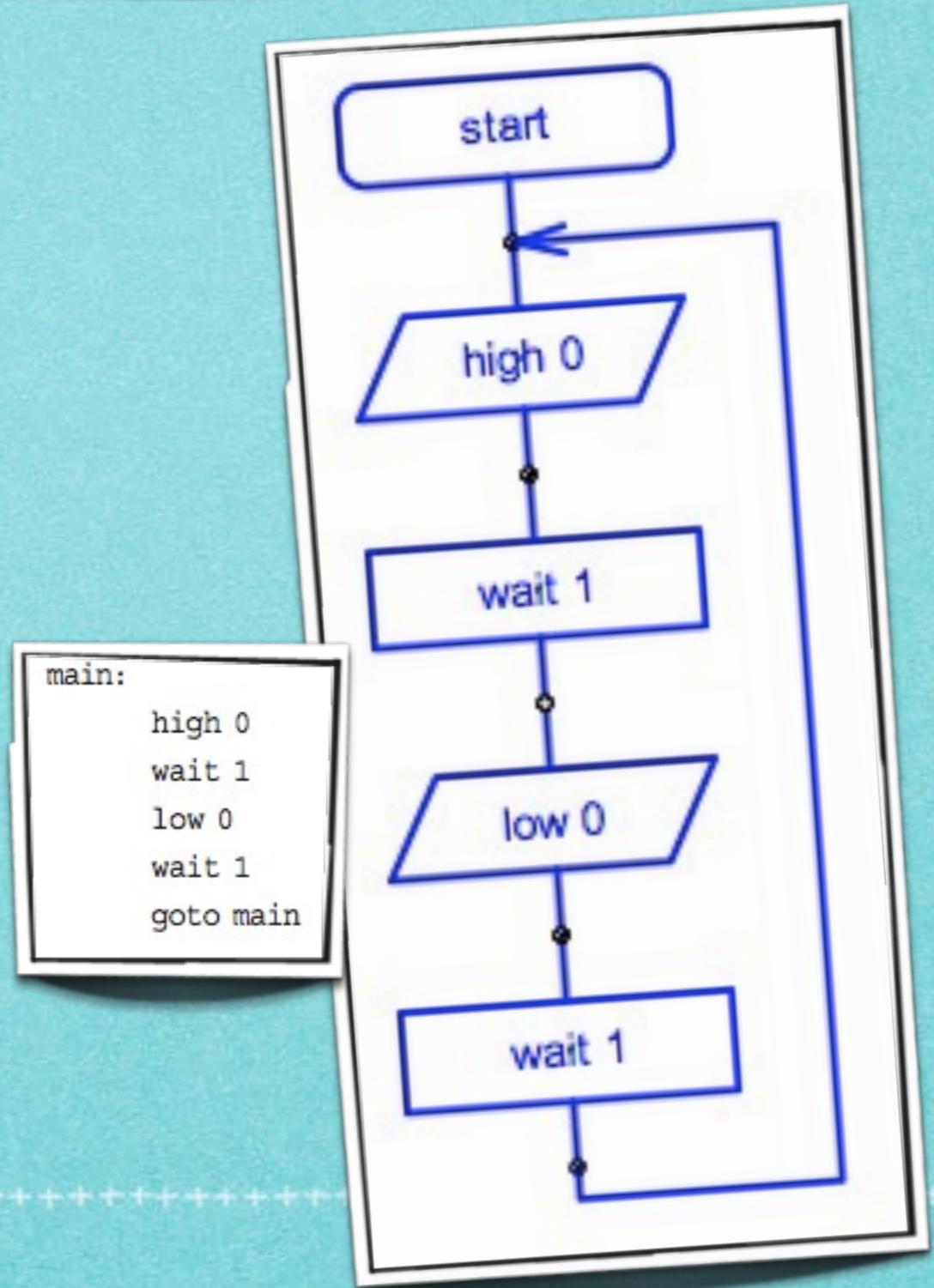
Blockdiagramme



- ▶ Blockdiagramme zeigen die Baugruppen einer elektronischen Schaltung
Eingabe ... Verarbeiten ... Ausgabe

Was ist das PICAXE-System?

- ▶ Microcontroller sind schwer zu programmieren:
 - ⦿ Schwierige Programmiersprache
 - ⦿ Spezielles Programmiergerät
- ▶ PICAXE:
 - ⦿ Flußdiagramme zeichnen
 - ⦿ Programmierung in BASIC
 - ⦿ Kein extra Programmiergerät

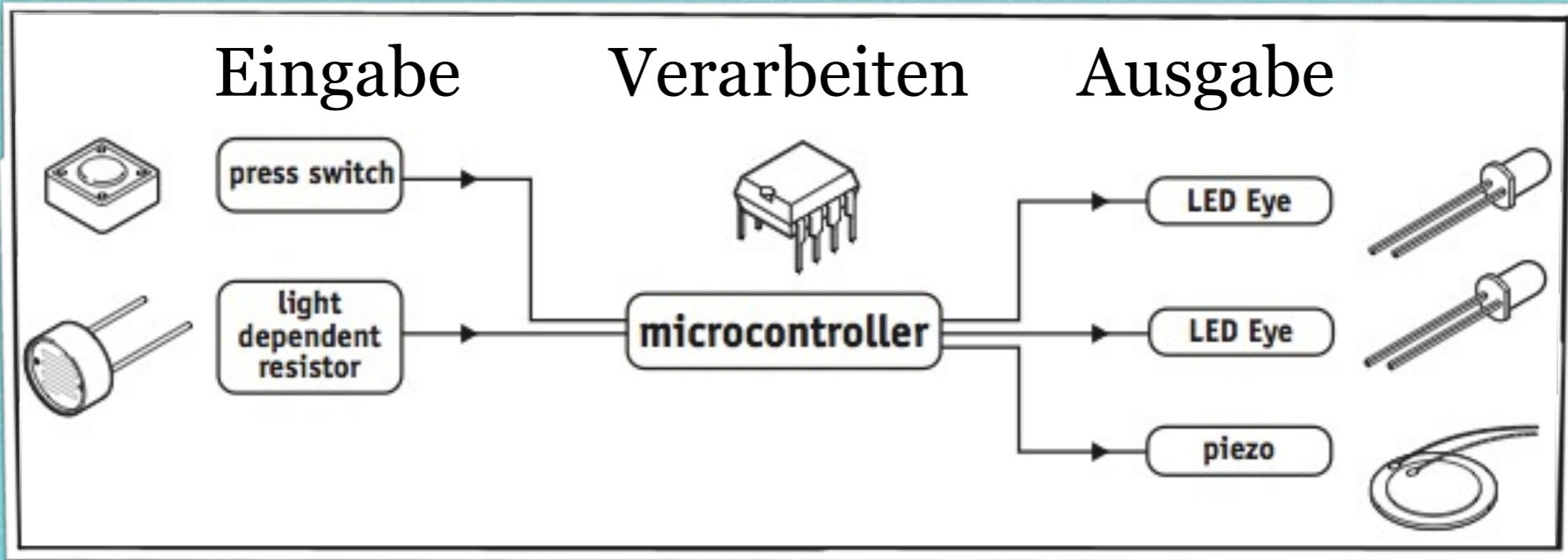


Wir bauen ein elektronisches Kuscheltier

- ▶ Gehirn: ein PICAXE-08
- ▶ LEDs als Augen, Piezo-Wandler für Töne, vielleicht Motor
- ▶ Reagiert auf Berührung, hell/dunkel
- ▶ Gestaltung
 - ⦿ flach - Pappe, Holz, Textil
 - ⦿ 3d - Drahtmodell, Stoffpuppe



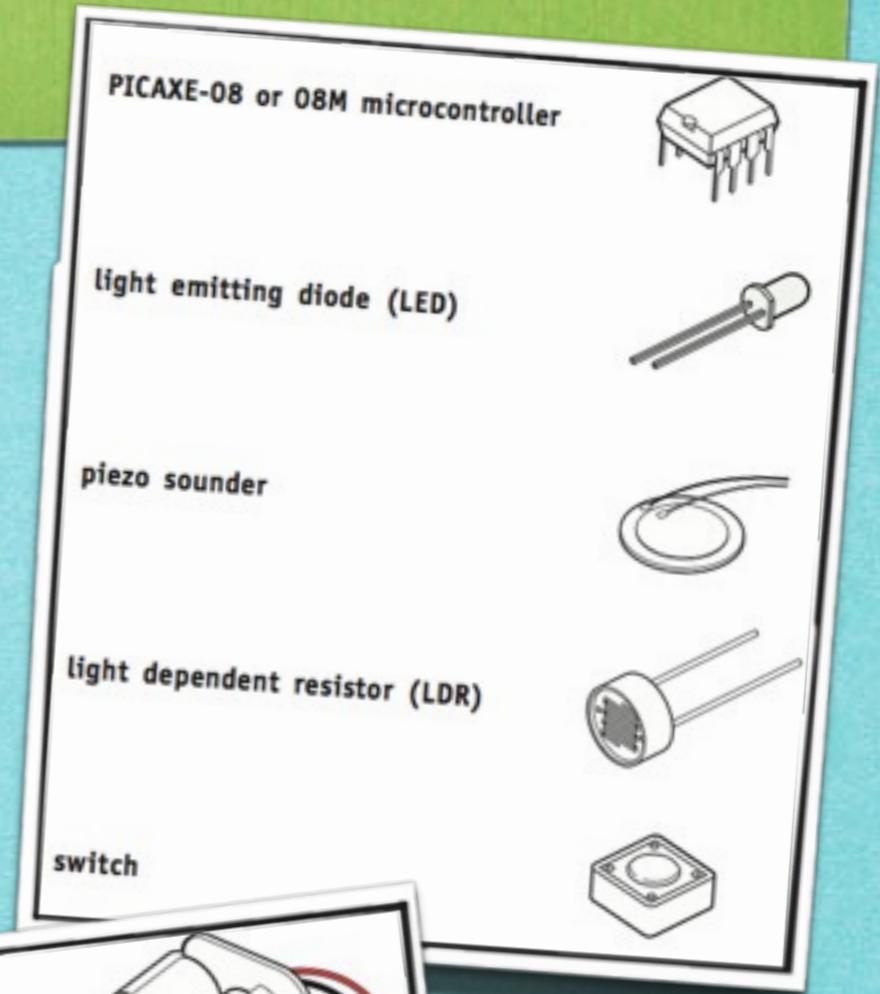
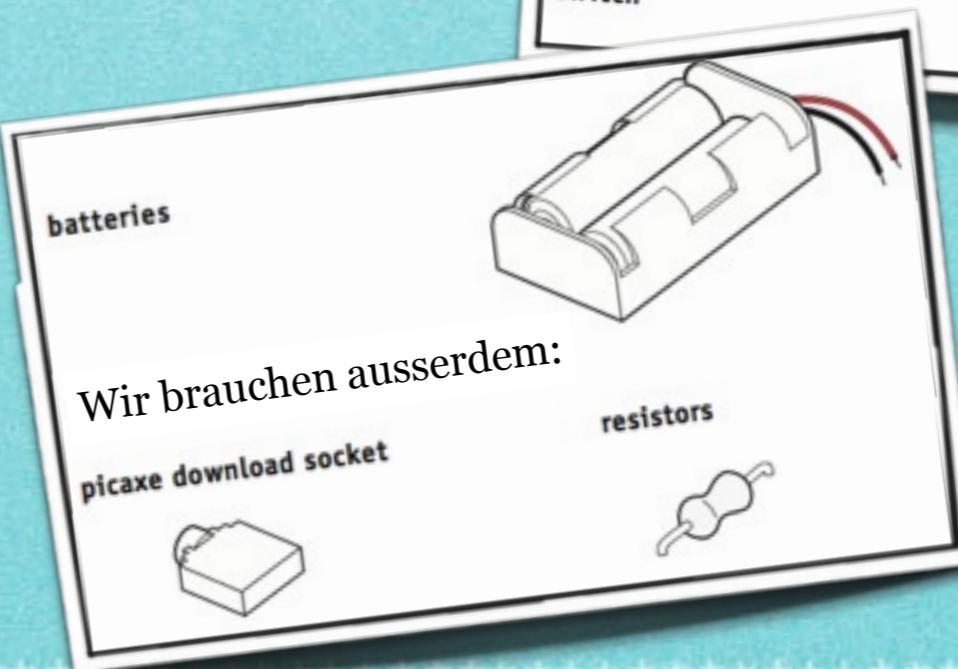
Blockdiagramm unseres Kuscheltiers



- ▶ Daran müssen wir denken
 - ⦿ Die Elektronik muss innen oder unter dem Kuscheltier eingebaut sein
 - ⦿ Die LEDs und der Lichtfühler sollen durch Löcher nach außen ragen
 - ⦿ Wir müssen später die Batterien auswechseln können
 - ⦿ Wo sollen die Verbindungsdrähte entlang geführt werden?

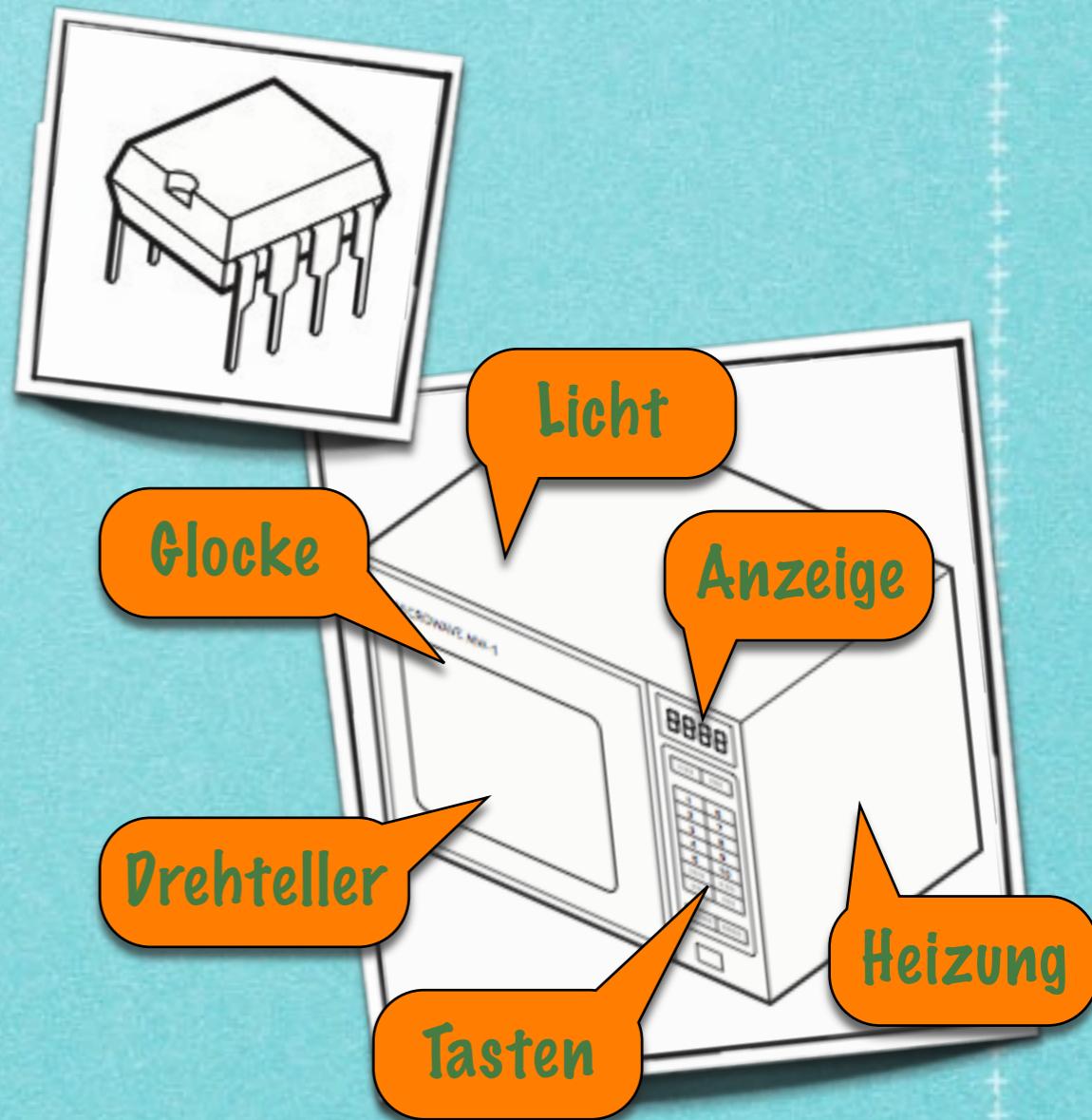
Elektronische Bauteile

- ▶ Auf den nächsten Seiten erfahren wir mehr über diese Bauteile ...

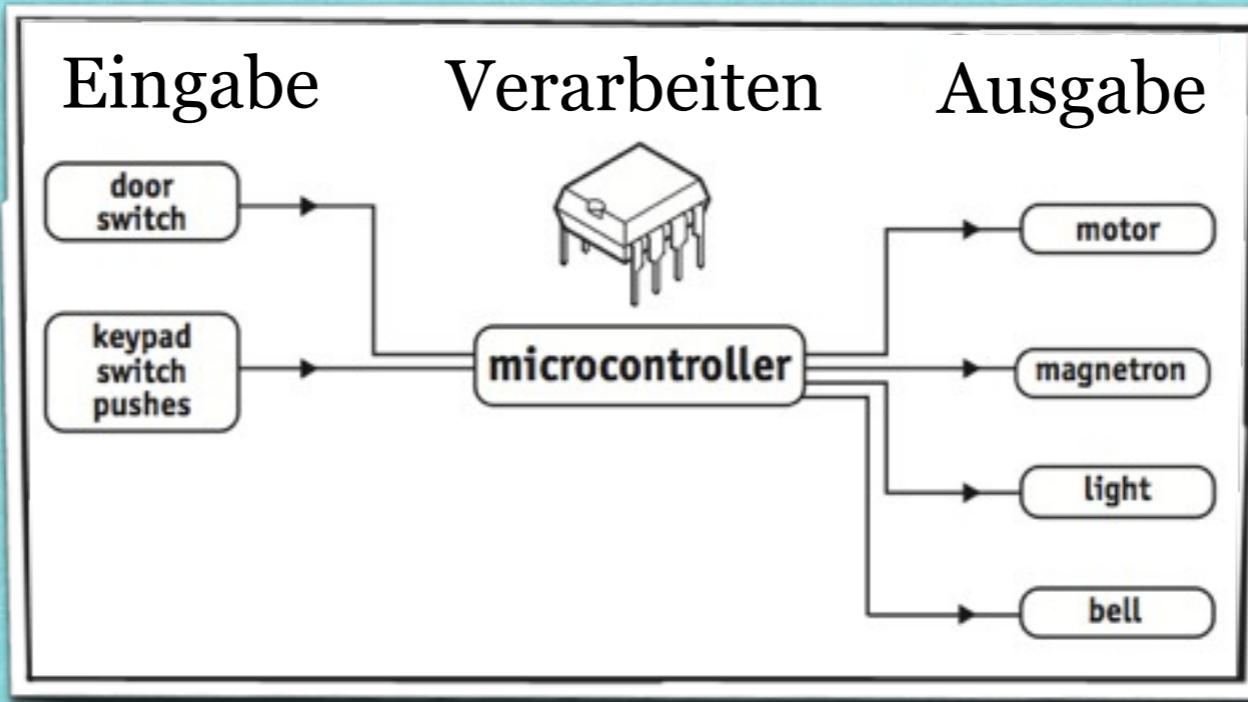


Microcontroller

- ▶ Computer auf einem Chip
- ▶ Werden nach der Programmierung in Produkte eingebaut, damit diese intelligenter und einfacher zu nutzen sind
- ▶ Beispiele:
 - ⦿ Haushaltsgeräte
 - ⦿ Alarmanlagen
 - ⦿ Medizinische Geräte
 - ⦿ in Fahrzeugen
 - ⦿ Messgeräte

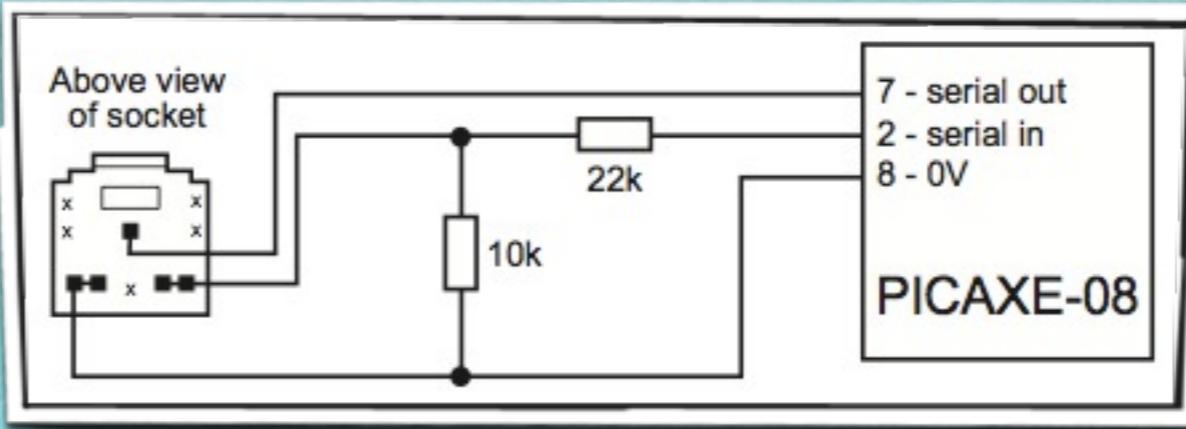


Wie kommen Programme auf den Microcontroller?



- ▶ Die Schaltung wird oft als Blockdiagramm gezeichnet
- ▶ Das Programm wird auf einen PC entwickelt
- ▶ Programme werden als Flußdiagramm gezeichnet oder in BASIC geschrieben
- ▶ ... und auf den Microcontroller hochgeladen

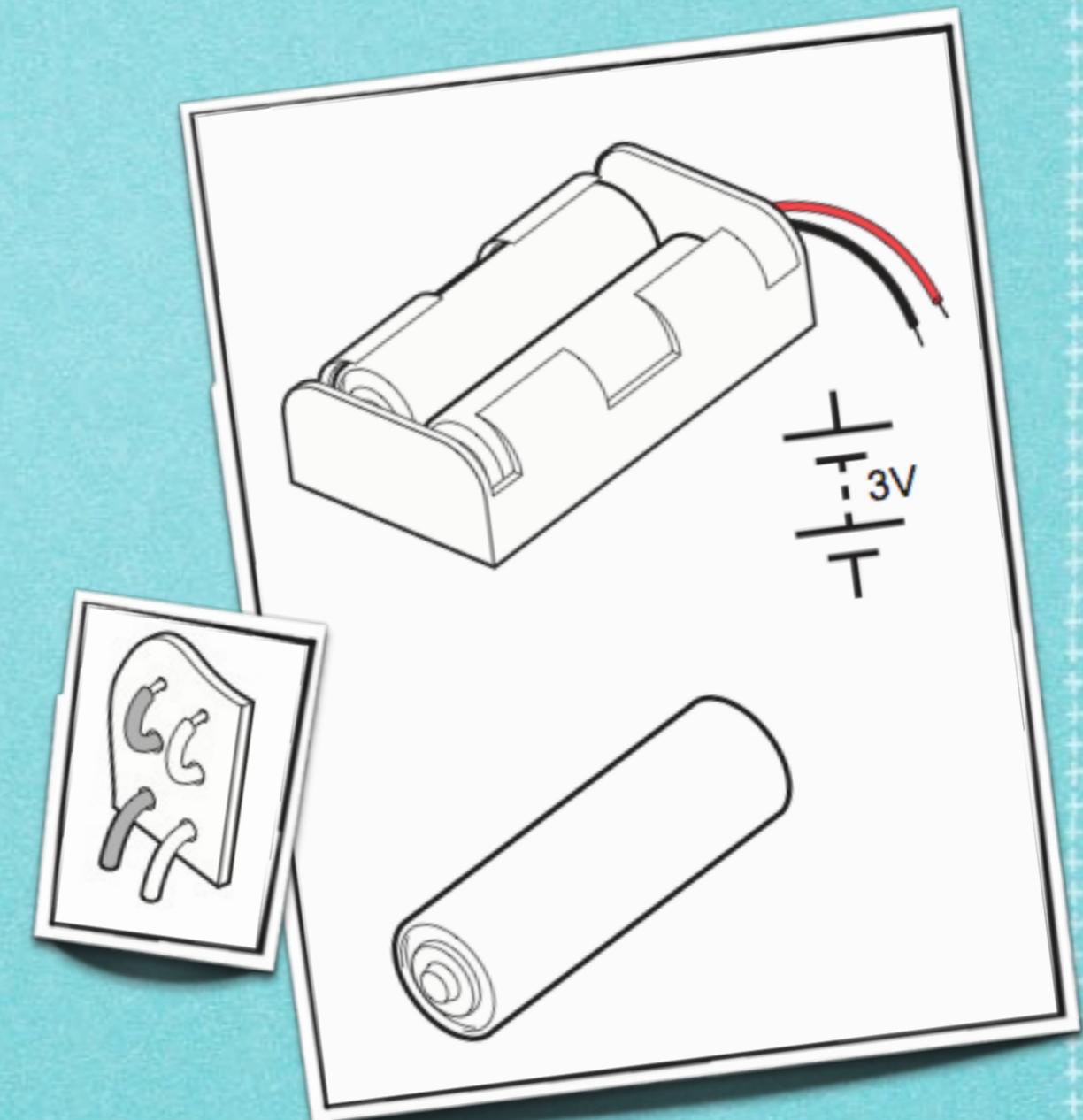
Wie werden Programme hochgeladen?



- ▶ Der PICAXE-08 Microcontroller wird über ein Kabel programmiert, das vom USB-Anschluß des PCs zur Platine mit dem Microcontroller verbunden wird
- ▶ Die Buchse auf der Platine (sieht aus wie ein Stereo-Kopfhöreranschluß) ist mit zwei Pins des Microcontrollers und dem Minuspol der Batterie verbunden
- ▶ Microcontroller und PC reden über dieses Kabel. So lädt der PC ein Programm in den Speicher des Microcontrollers

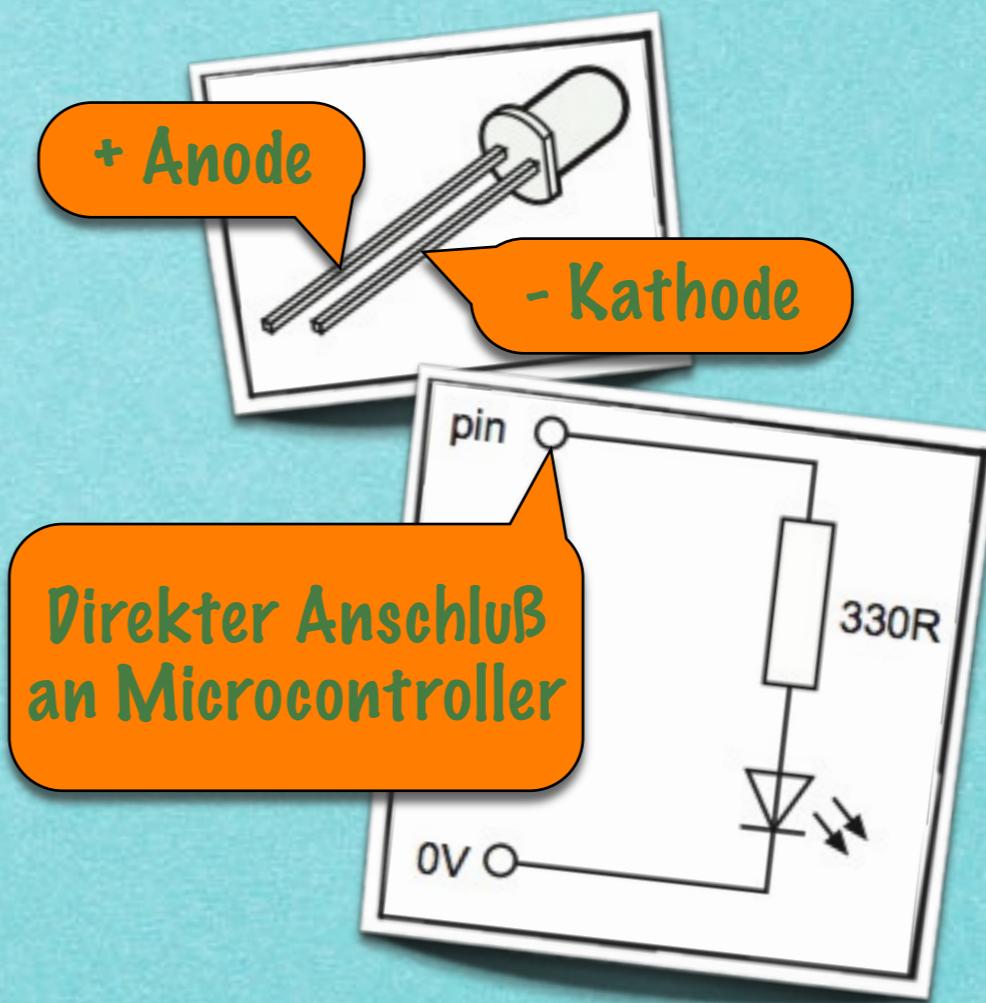
Batterien

- ▶ Wandeln chemische in elektrische Energie um
- ▶ 3 Zellen á 1,5V = 4,5V
- ▶ Microcontroller benötigen 3...5V, daher funktionieren 3 AA-Zellen gut.
- ▶ Niemals 9V-Blockbatterien benutzen!
- ▶ Batterien niemals kurzschließen!
- ▶ Auf die Polung achten:
+ rot / - schwarz



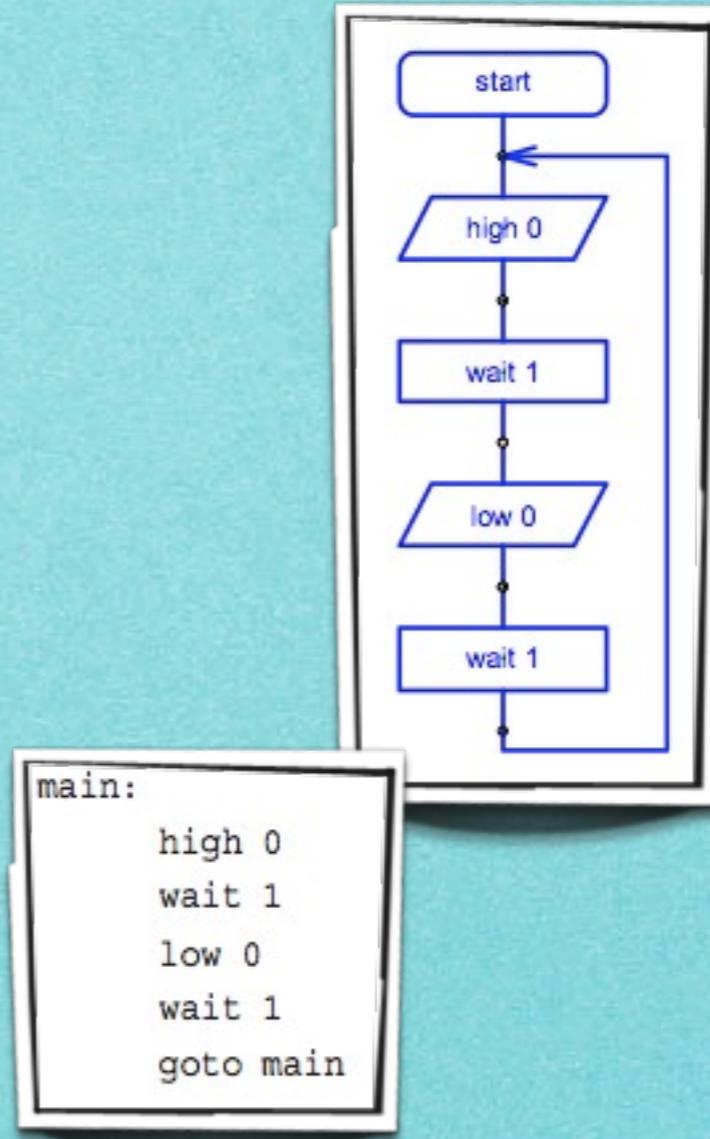
LEDs - Leuchtdioden

- ▶ Spezielle Dioden, die beim Anlegen von Strom Licht erzeugen
- ▶ Diode = Strom fließt nur in einer Richtung
- ▶ Benutzt als Anzeige und immer öfter zur Beleuchtung
- ▶ Infrarot-LEDs stecken in Fernsteuerungen
- ▶ LEDs brauchen einen Vorwiderstand:
 - ⦿ 4,5V Batterie => 330 Ohm
 - ⦿ 3V Batterie => 120 Ohm



Test einer LED-Verbindung

- ▶ Dieses einfache Programm schaltet die LED an Ausgangs-Pin 0 jede Sekunde an und aus
- ▶ Falls die die LED nicht blinkt:
 - ⌚ Ist die LED richtig herum angeschlossen?
 - ⌚ Ist der richtige Widerstand eingebaut?
 - ⌚ Wurde der richtige Ausgangs-Pin benutzt?
 - ⌚ Stimmen alle Verbindungen?



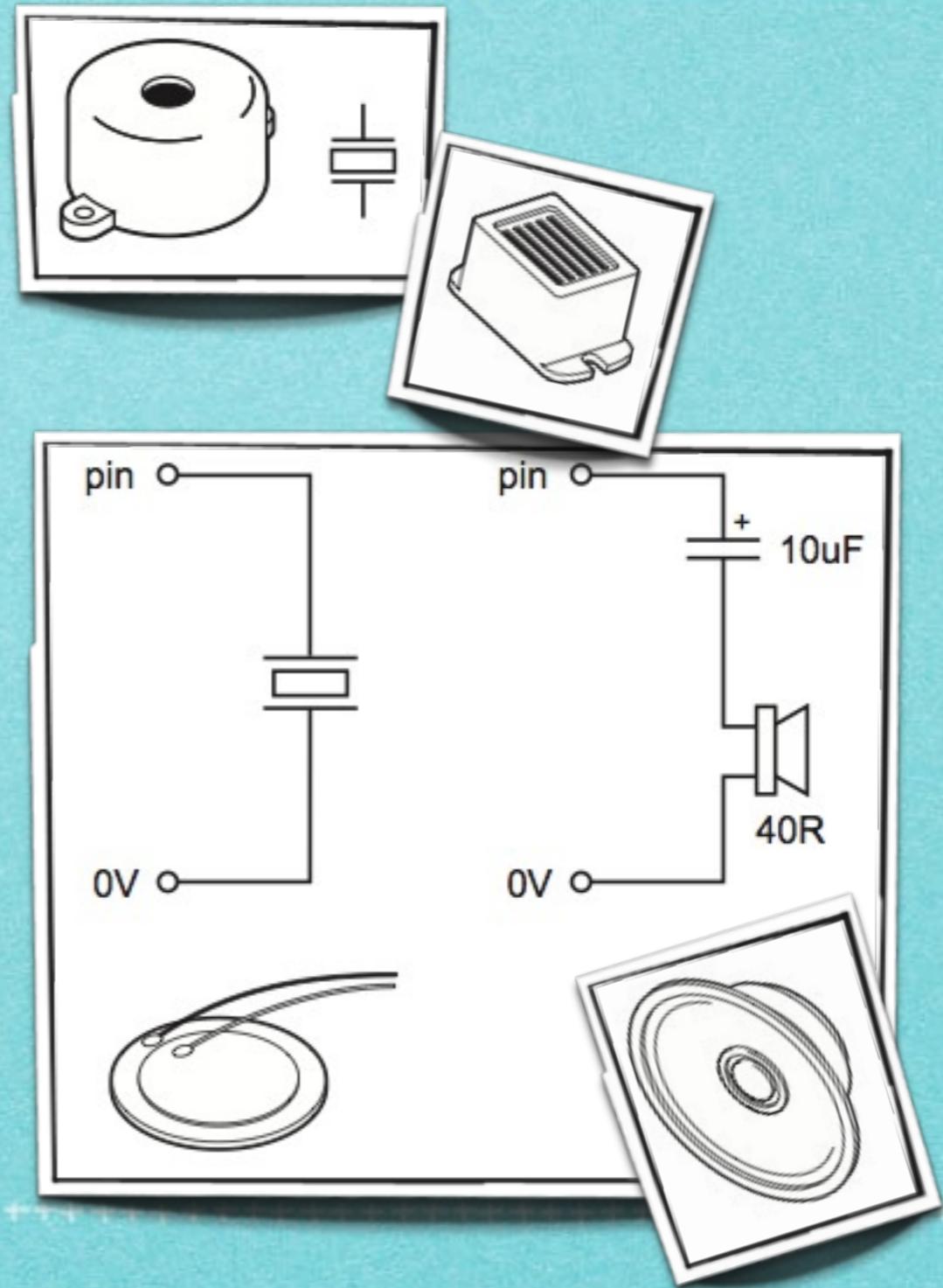
Ein zweites LED-Testprogramm - 15x an/aus

- ▶ Dieses Programm schaltet die LED an Pin 0 mit einer **for...next-Schleife** 15x an und aus
- ▶ Die Anzahl wird in der **Speicher-Variable b1** gemerkt. Es gibt 14 solcher Variablen bo .. b13
- ▶ Variablen sind Zahlen-speicher im Microcontroller

```
main: for b1 = 1 to 15    ' start a for...next loop
        high 0           ' switch pin 0 high
        pause 500         ' wait for 0.5 second
        low 0            ' switch pin 0 low
        pause 500         ' wait for 0.5 second
    next b1             ' end of for...next loop
end                   ' end program
```

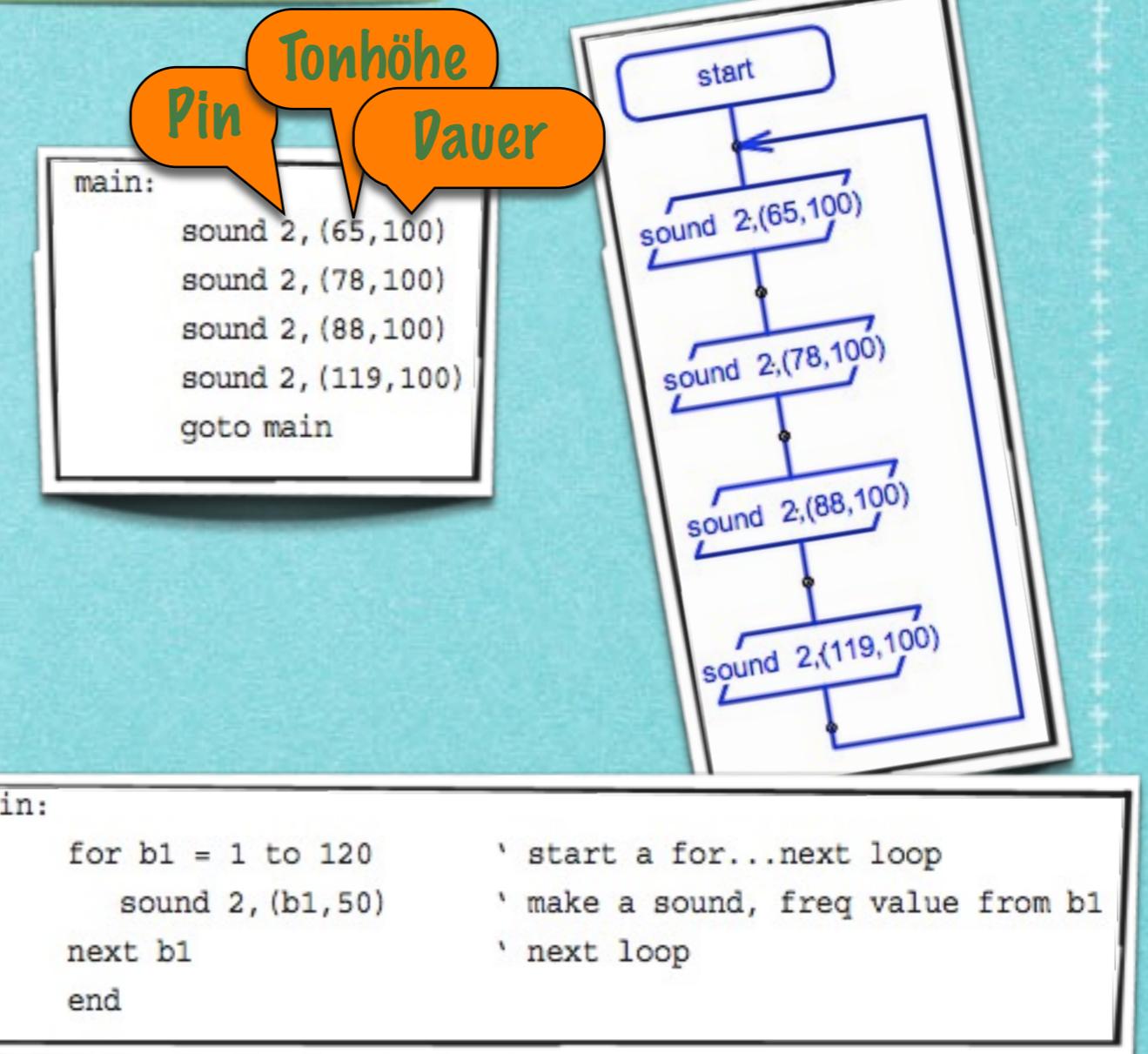
Piezo-Schallwandler

- ▶ Piezo-Schallwandler sind preiswerte Mini-Lautsprecher
- ▶ Einsatz zur Rückmeldung an Nutzer oder in klingenden Geburtstagskarten
- ▶ Piezos können direkt mit einem Microcontroller-Pin verbunden werden
- ▶ Für lautere Klänge kann ein Lautsprecher mit Kondensator benutzt werden



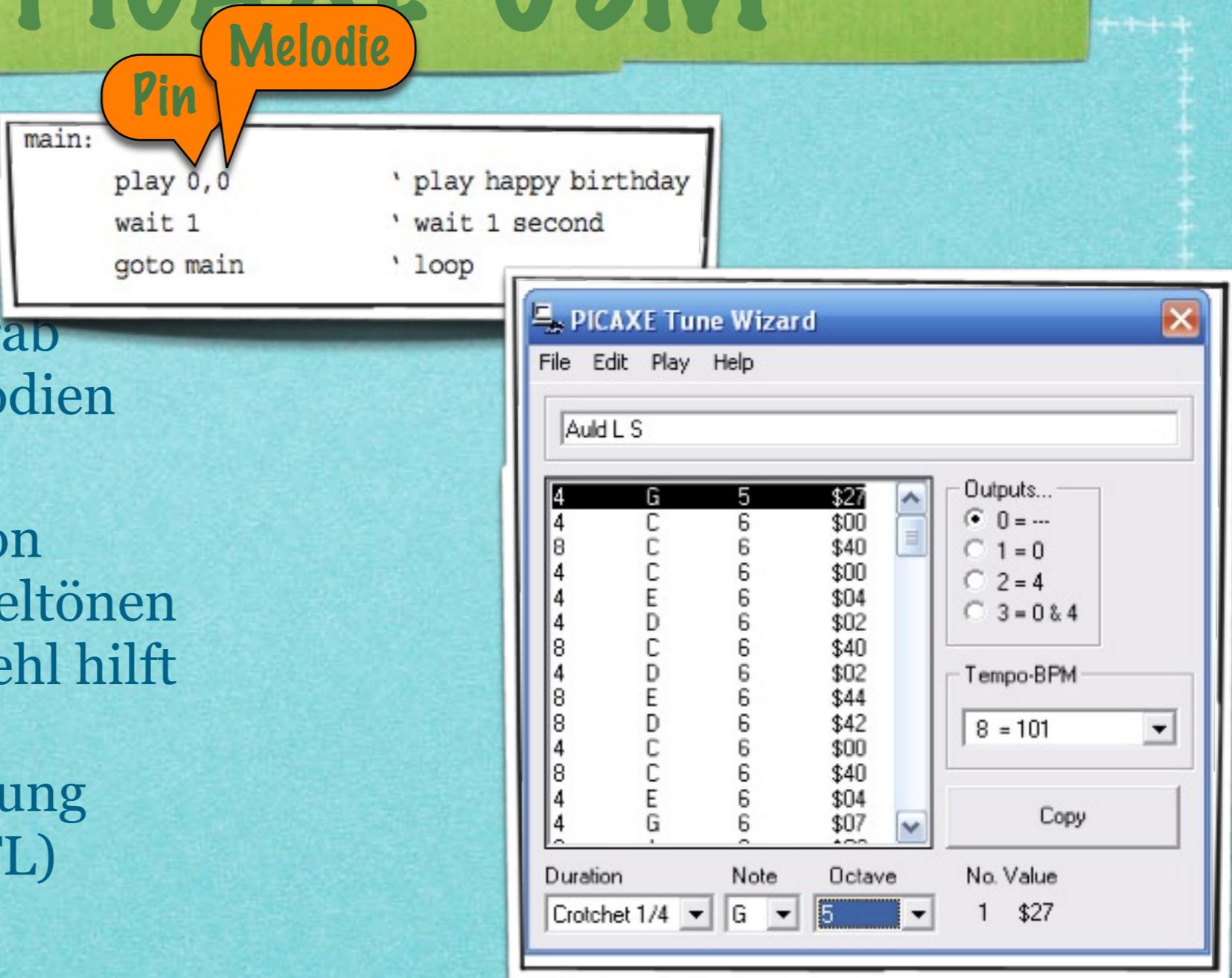
Test der Piezo-Verbindung

- ▶ Dieses einfache Programm lässt den Piezo 4 verschiedene Töne machen
- ▶ Falls es nicht funktioniert:
 - ⦿ Ist die Tonhöhe zwischen 0 und 127?
 - ⦿ Wurde der richtige Pin benutzt?
 - ⦿ Stimmen alle Verbindungen?

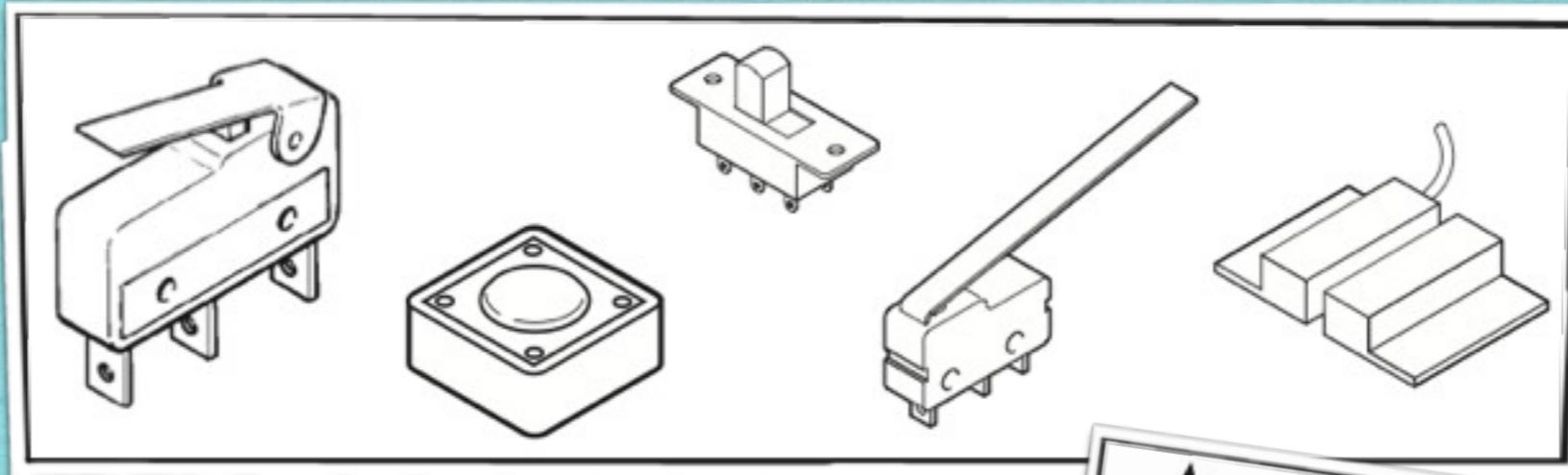


Spielen von Melodien auf dem PICAXE-08M

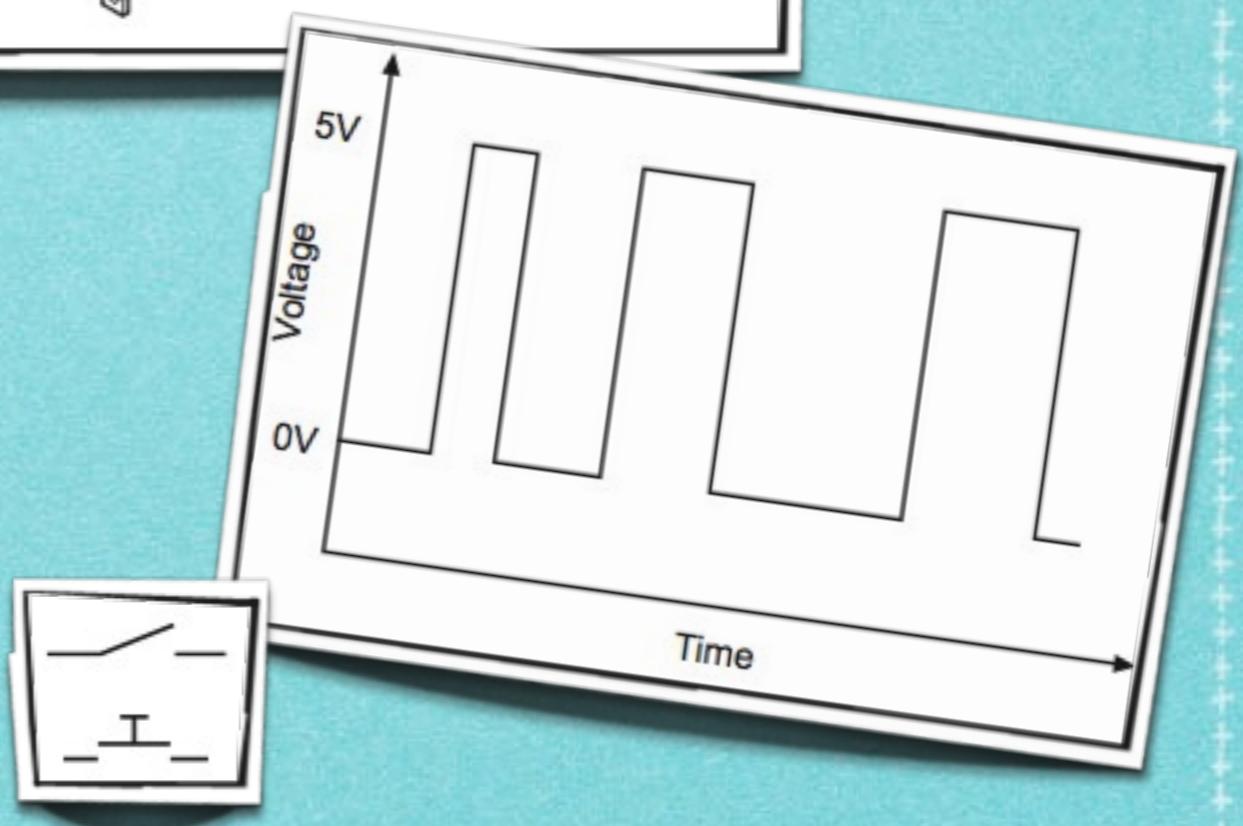
- ▶ Der 08M kann 4 vorab programmierte Melodien spielen
- ▶ Für das Abspielen von Nokia-RTTTL-Klingeltönen über den „tune“-Befehl hilft ein Assistent in der Programmierumgebung (File > Import RTTTL)



Digitale Fühler (Schalter)



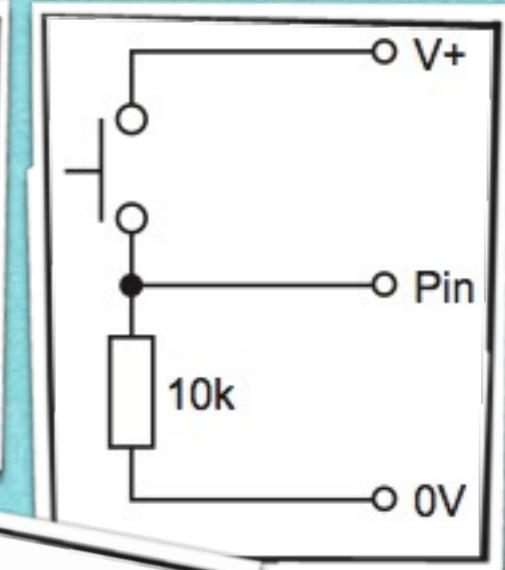
- ▶ Drucktaster, Microschalter,
Reed-Schalter, Kippschalter
- ▶ Anwendung in Tastaturen,
Alarmanlagen,
Türschließung,
Bewegungsfühler



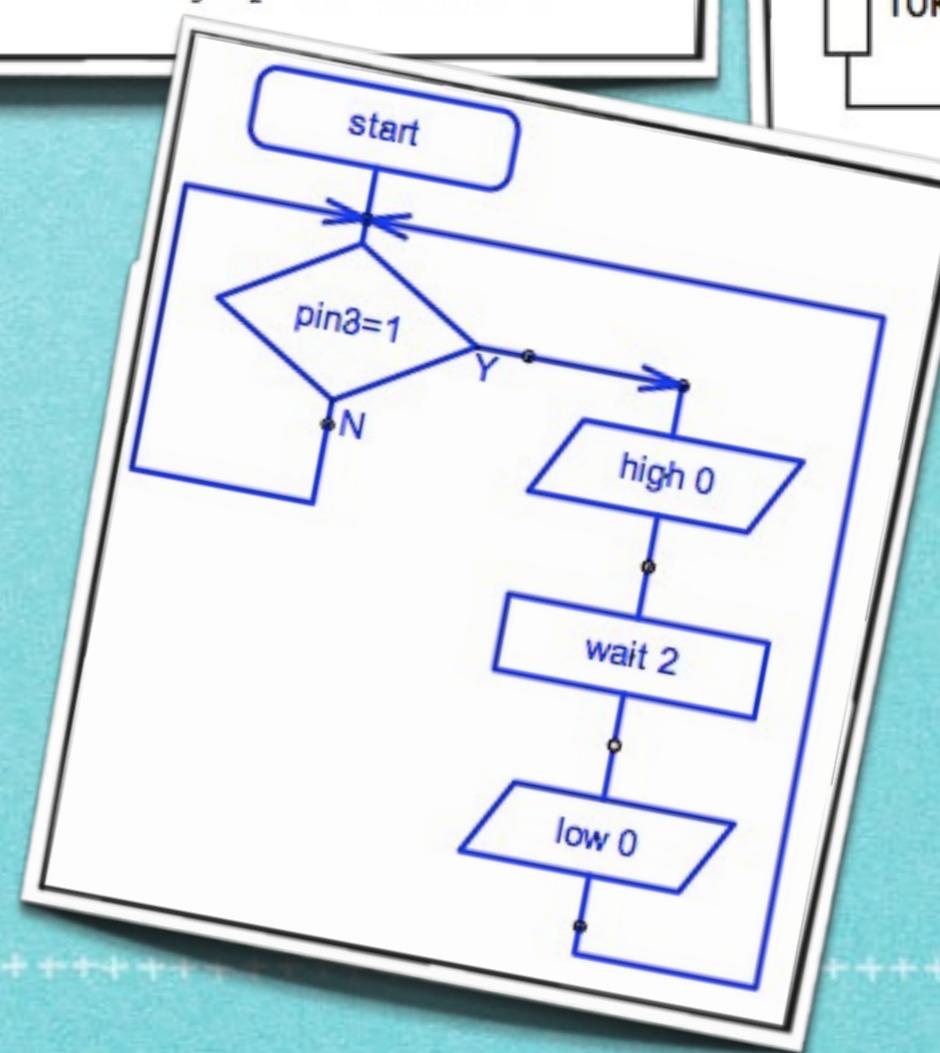
Anwendung von Schaltern

```
main:  
    if input3 = 1 then  
        high 0  
        wait 2  
        low 0  
    end if  
    goto main
```

```
' make a label called 'main'  
' test if the input is on  
' switch output 0 on  
' wait 2 seconds  
' switch output 0 off  
  
' jump back to start
```

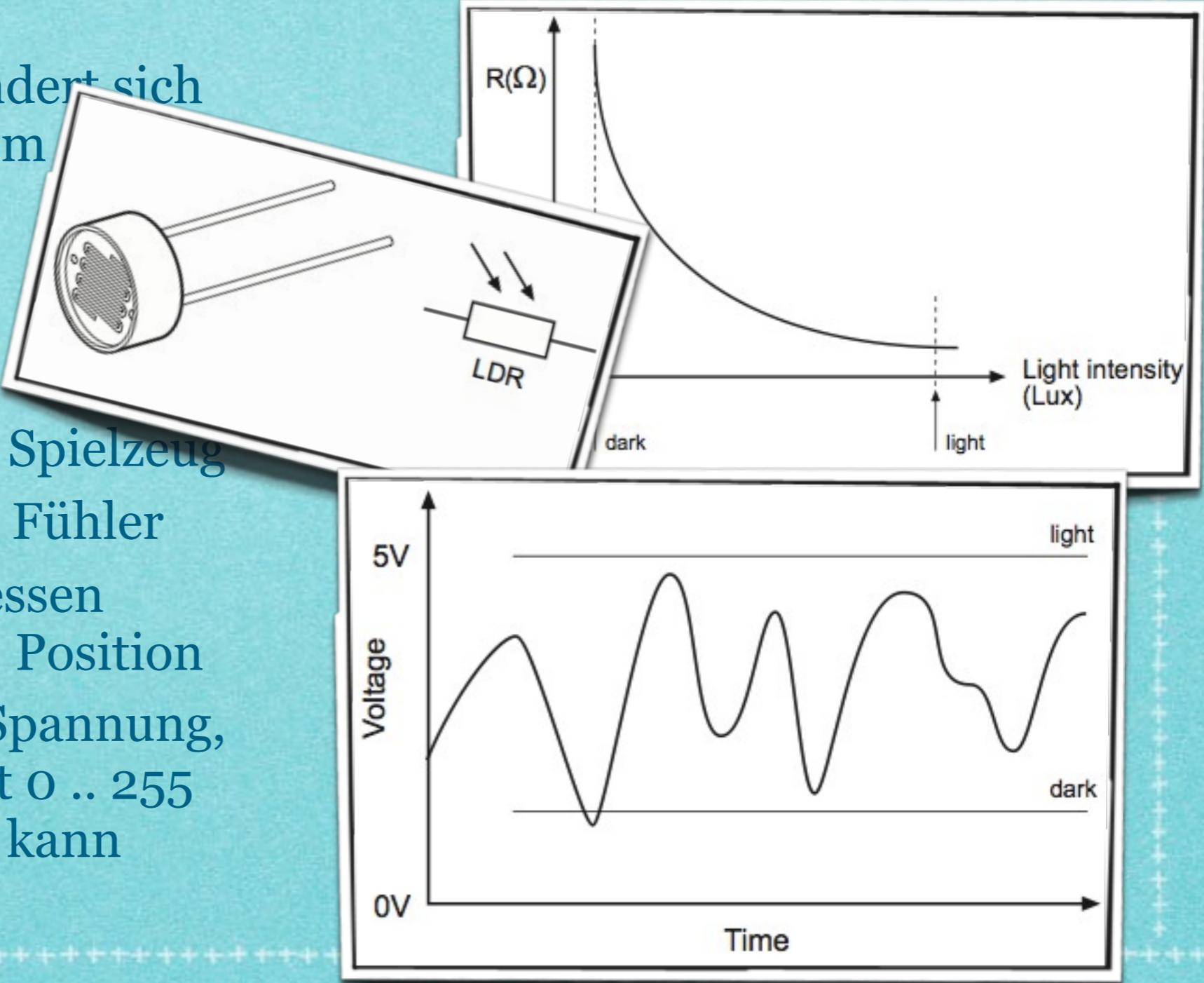


- ▶ Schaltung mit Widerstand
- ▶ Das Programm schaltet einen Ausgang an und aus, wenn der Schalter an Eingang 3 geschlossen ist



Lichtempfindliche Widerstände (LDR)

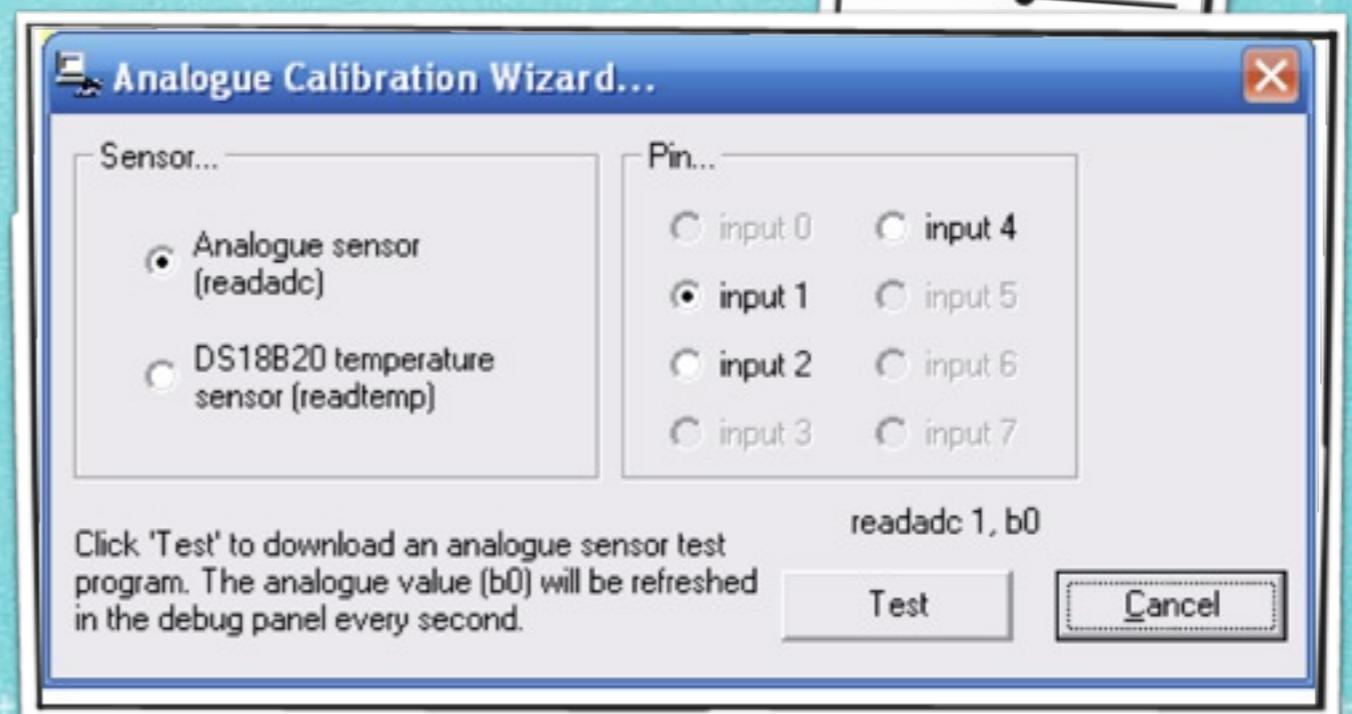
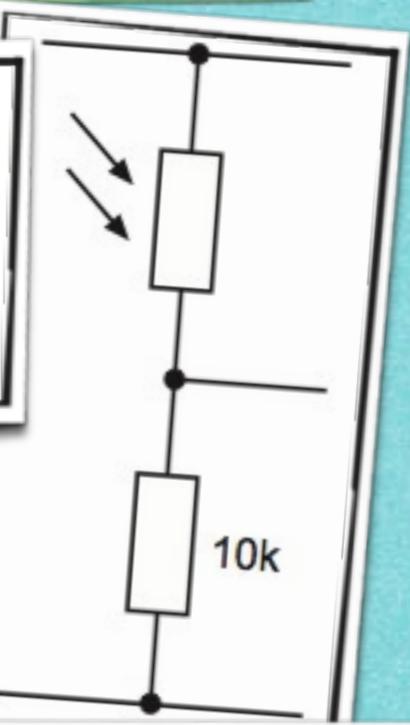
- ▶ Der Widerstand ändert sich in Abhängigkeit vom Lichteinfall
- ▶ Anwendung in automatischen Straßenlaternen, Alarmanlagen und Spielzeug
- ▶ LDRs sind analoge Fühler
- ▶ Analoge Fühler messen Licht, Temperatur, Position
- ▶ Sie erzeugen eine Spannung, die mit einem Wert 0 .. 255 dargestellt werden kann



Anwendung von LDRs

- ▶ LDRs können als Schalter oder als Analog-Fühler benutzt werden
- ▶ An den PICAXE-08-Microcontrollern kann **Pin1** analoge Signale messen
- ▶ Ein Assistent in der Programmierumgebung hilft einen Analog-Fühler einzustellen

```
main:          ' make a label called main
               readadc 1,b1      ' read channel 1 into variable
               debug b0         ' transmit value to computer
               pause 1000       ' short delay
               goto main        ' jump back to the start
```



Test des LDRs digital/analog

Pin Variable

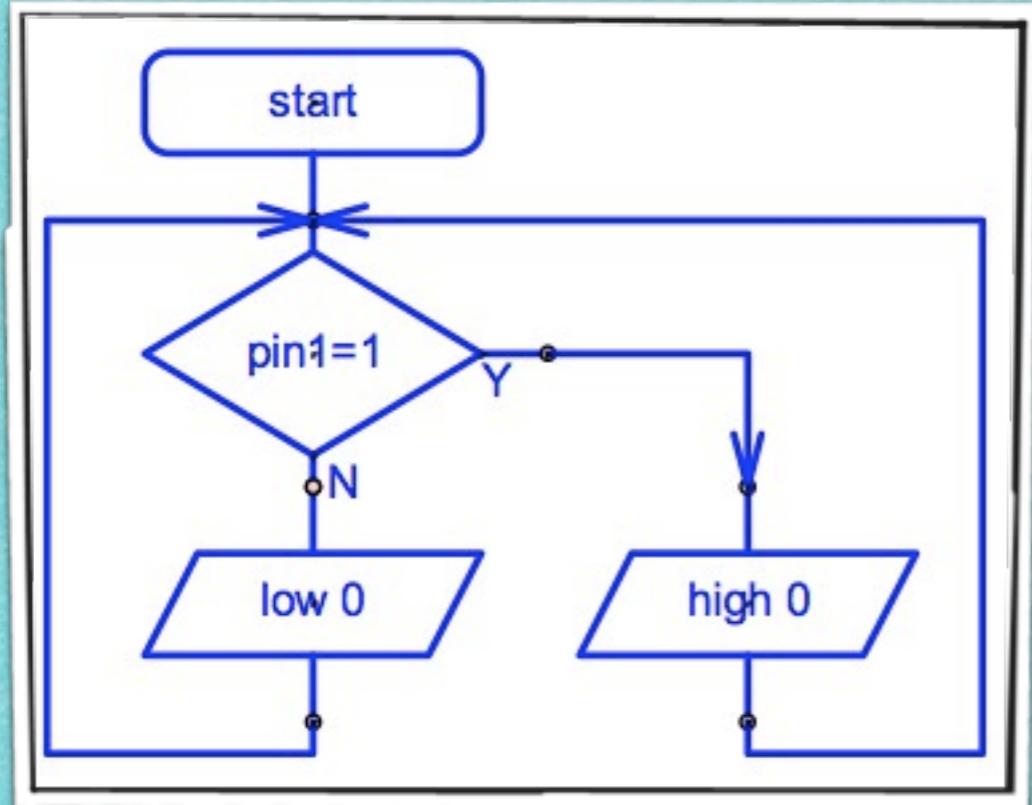
```

main:
readadc 1,b1
if b1 > 100 then do4
if b1 > 50 then do0
low 0
low 4
goto main

do4:
high 4
low 0
goto main

do0:
high 0
low 4
goto main

```

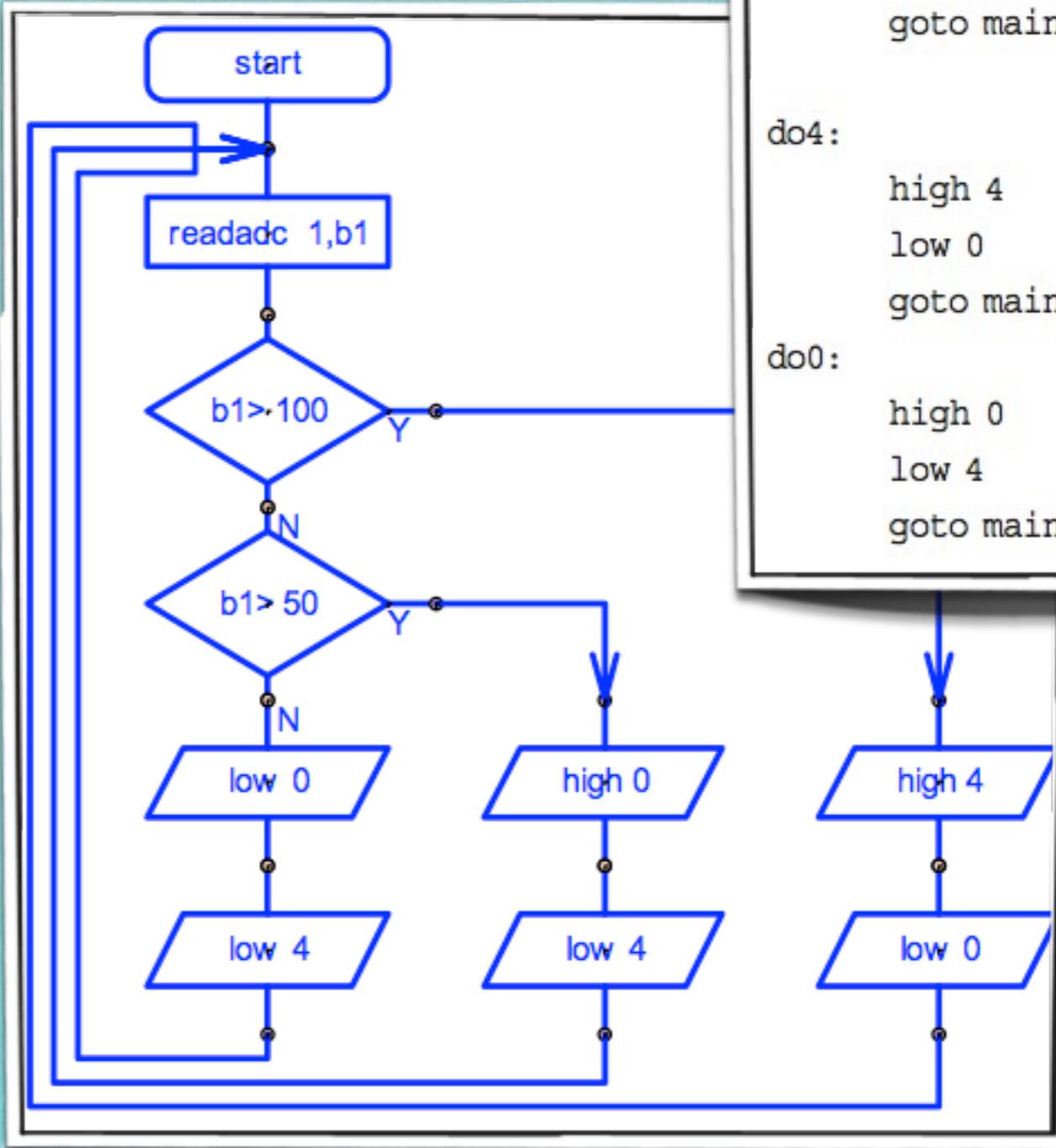


```

main:
if input1 is on then dohigh
low 0
goto main

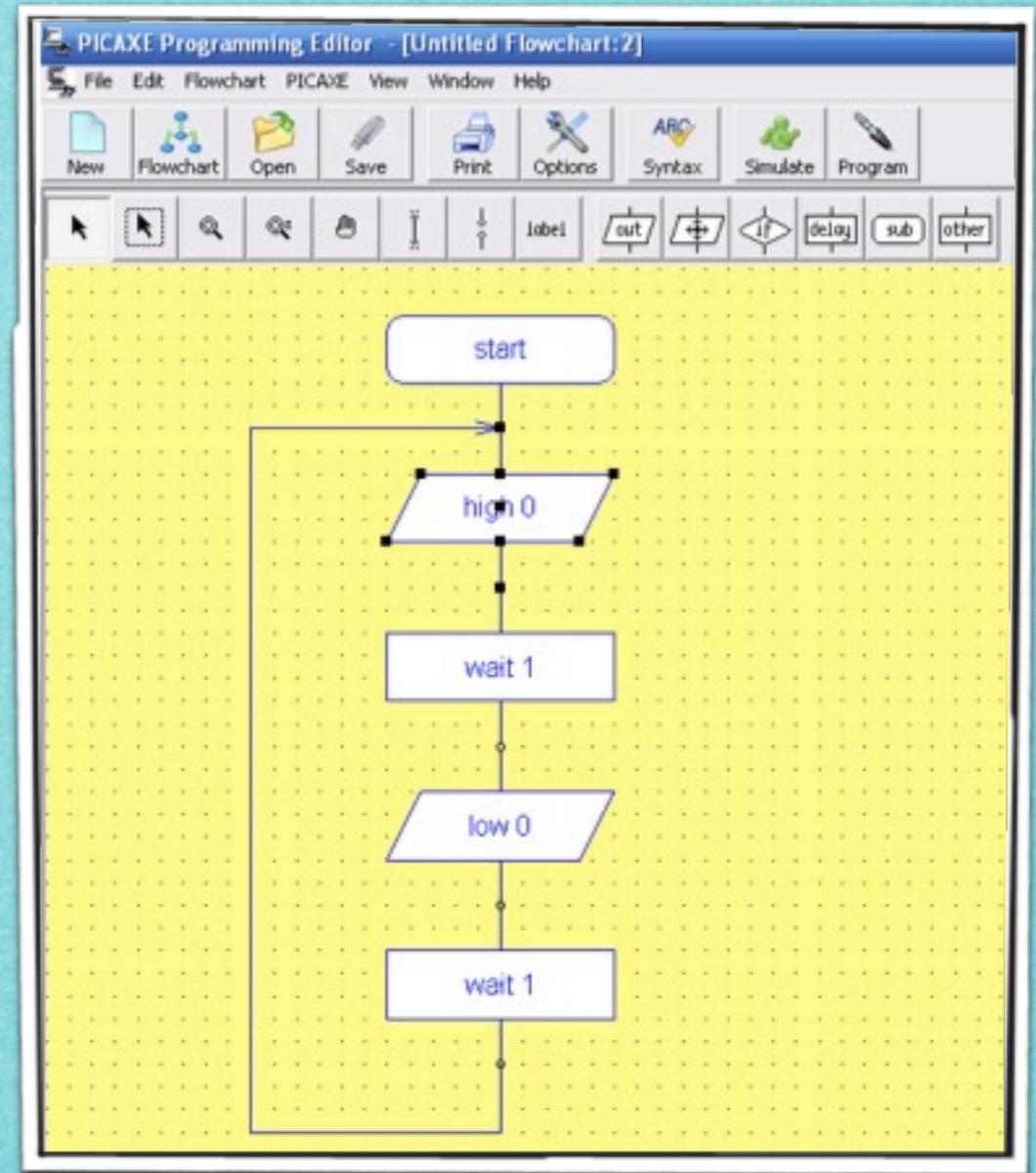
dohigh:
high 0
goto main

```

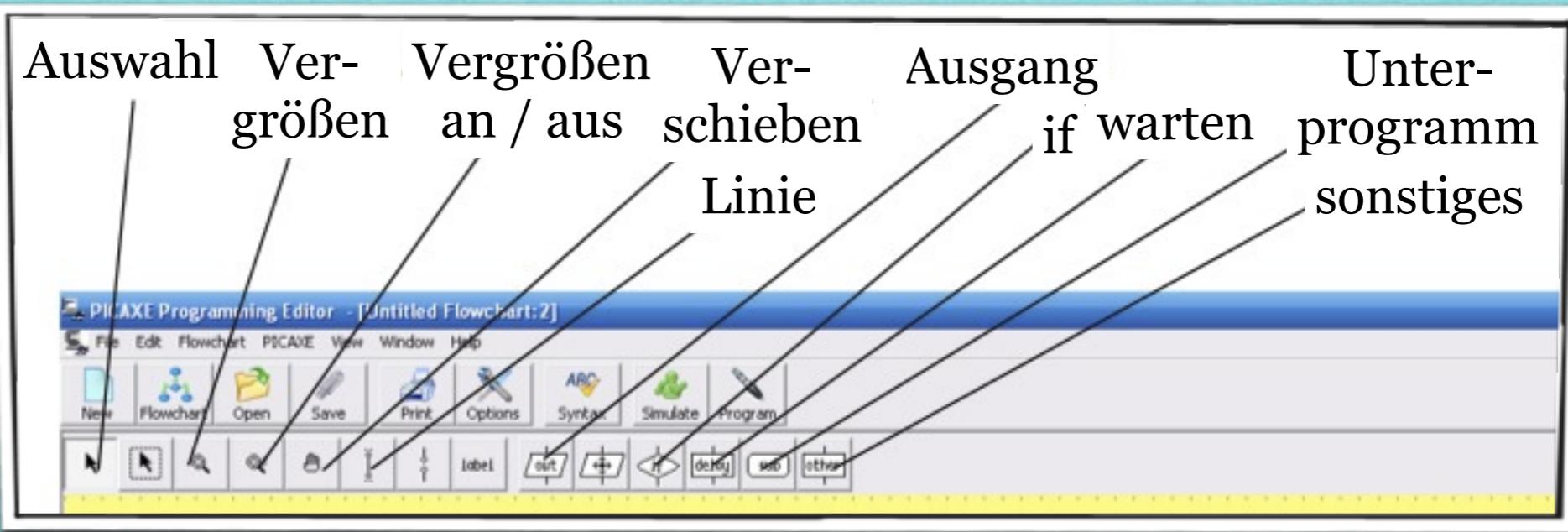


Flussdiagramme

- ▶ Flußdiagramme helfen Programme zu verstehen:
 - ⦿ Verbinde das Kabel
 - ⦿ Starte die Programmierumgebung
 - ⦿ Wähle unter View > Options im Mode-Reiter „08M“
 - ⦿ Wähle den seriellen Anschluß im Serial Port-Reiter
 - ⦿ Beginne ein neues Flußdiagramm mit File > New Flowchart
 - ⦿ Zeichne das Flußdiagramm
 - ⦿ Wandle es über Flowchart > Convert Flowchart to BASIC
 - ⦿ Lade das erzeugte BASIC-Programm über PICAXE > Run auf den Microcontroller
 - ⦿ Drucken / Speichern / Export als Grafik erfolgt über das File-Menü



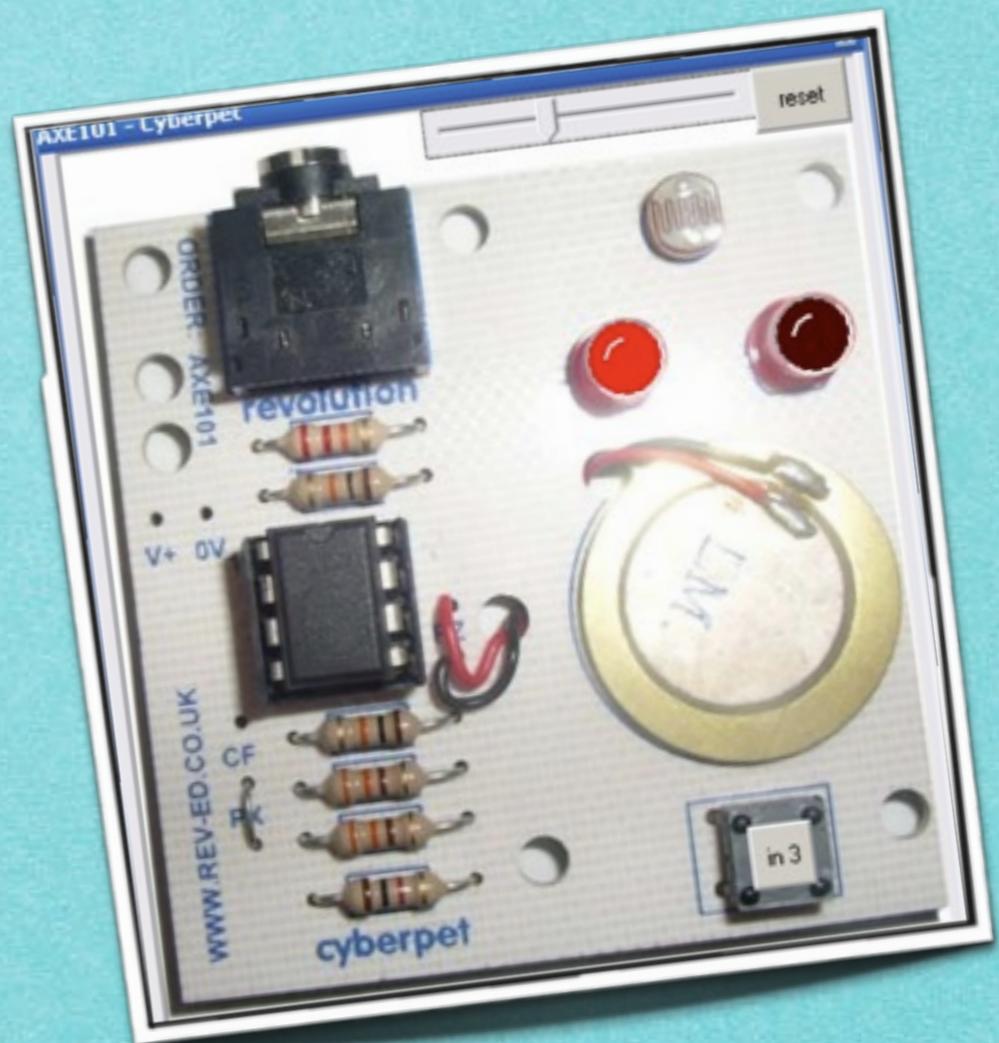
Flußdiagramme zeichnen



- ▶ Zeichnen eines Flußdiagramms:
 - ⦿ Klicke auf einen der Knöpfe: if, warten, Unterprogramm, sonstiges
 - ⦿ Wähle das einen Befehl aus dem Aufklappmenü
 - ⦿ Klicke auf die Arbeitsfläche, um das Befehlssymbol zu platzieren
 - ⦿ Bearbeite den zugehörigen BASIC-Befehl am unteren Fensterrand
 - ⦿ Symbole verbinden: Nah zueinander verschieben oder Linie zeichnen
 - ⦿ Ecken in Linien können durch Anklicken erzeugt werden

Programm simulieren

- ▶ Der Programmeditor bietet zwei Arten der Simulation:
 - ⦿ **Ablaufsimulation des Flußdiagramms:** Klicke „Simulate“ im Simulate-Menü
 - ⦿ Die Geschwindigkeit kann in View > Options > Flowchart Menu eingestellt werden
 - ⦿ Simulation mit Platine: Klicke „AXE 101 Cyberpet“ in Simulate > Simulation Panels > Product Sims
 - ⦿ Du kannst den Taster drücken und die Helligkeit über den Schieber einstellen. Die LEDs folgen den Ausgabebefehlen

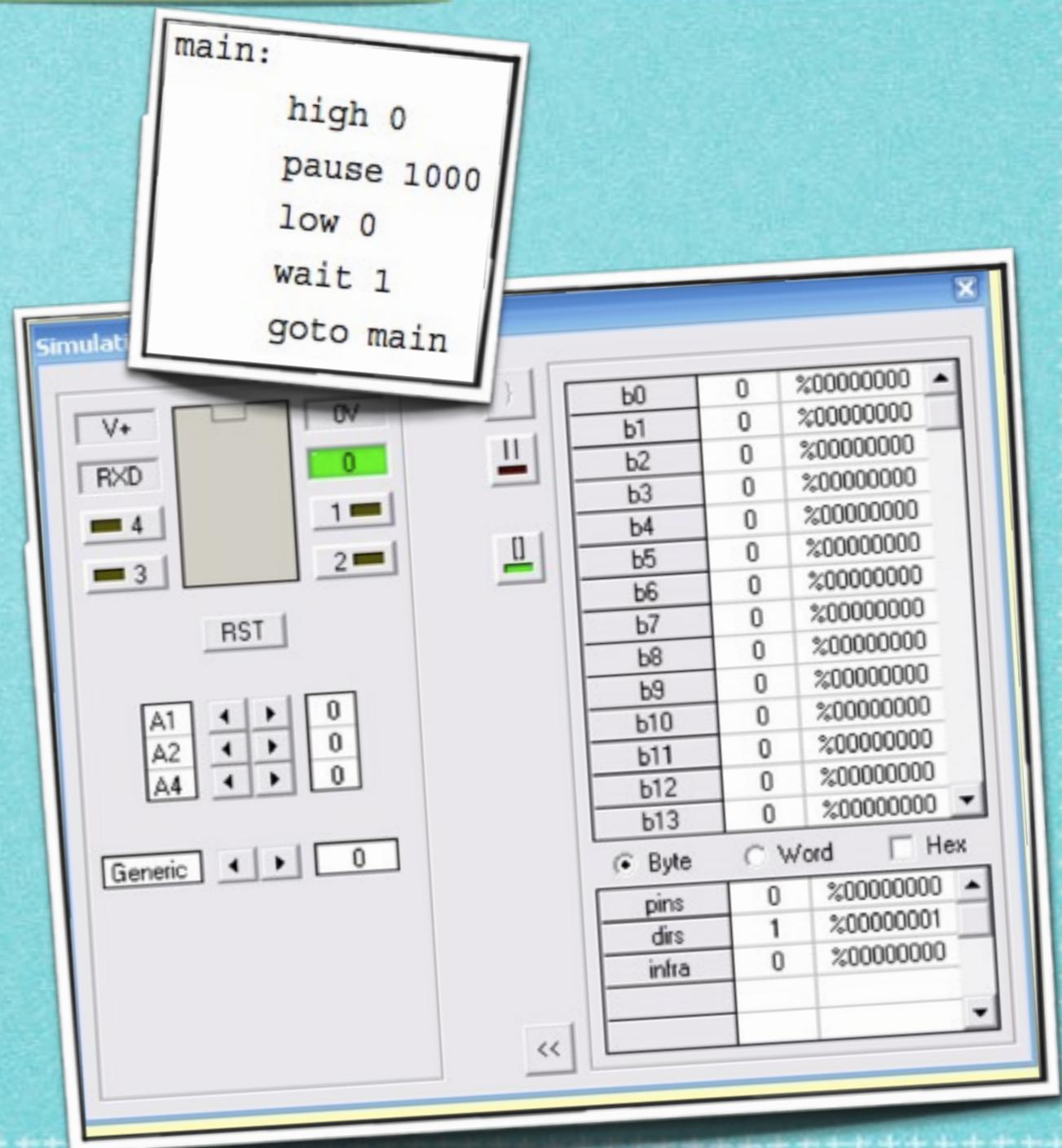


Flußdiagramme hochladen

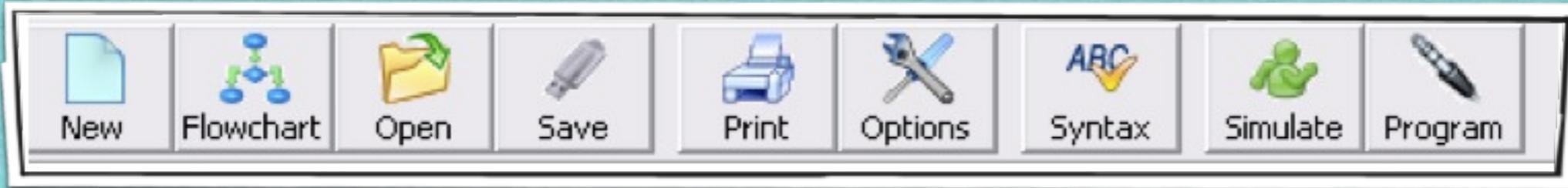
- ▶ Flußdiagramme können direkt auf den Microcontroller geladen werden:
 - ▶ Menüpunkt „Programm“
- ▶ Bitte beachten:
 - ▶ Das Wandeln nach BASIC bricht bei nicht verbundenen Symbole ab
 - ▶ Immer ein Stop-Symbol benutzen
 - ▶ Umwandeln / Herunterladen: 2x F5 drücken

Programmieren in BASIC

- ▶ BASIC hat mehr Befehle als Flußdiagramme (z.B. for .. next)
 - ⌚ Das einfache Programm rechts schaltet Ausgang 0 jede Sekunde an / aus
- ▶ BASIC-Programme können auch simuliert werden:
 - ⌚ Menüpunkt Simulate > Run

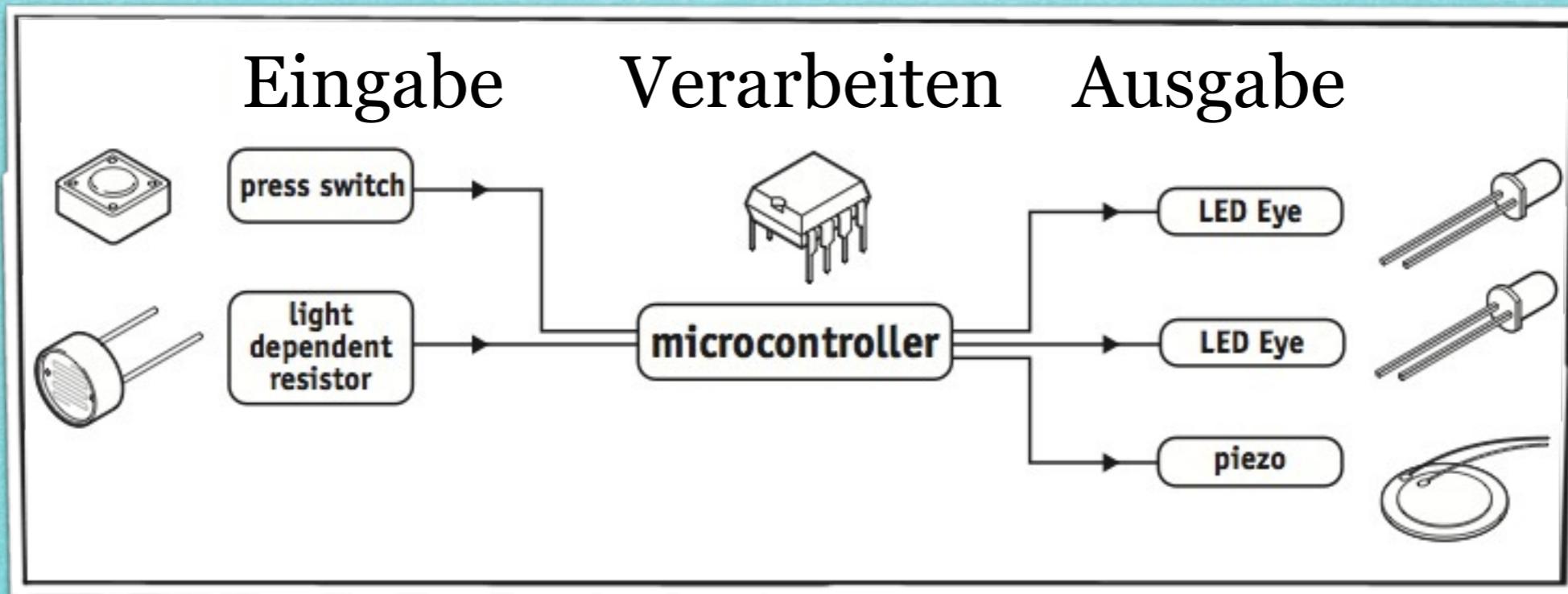


Programmierumgebung

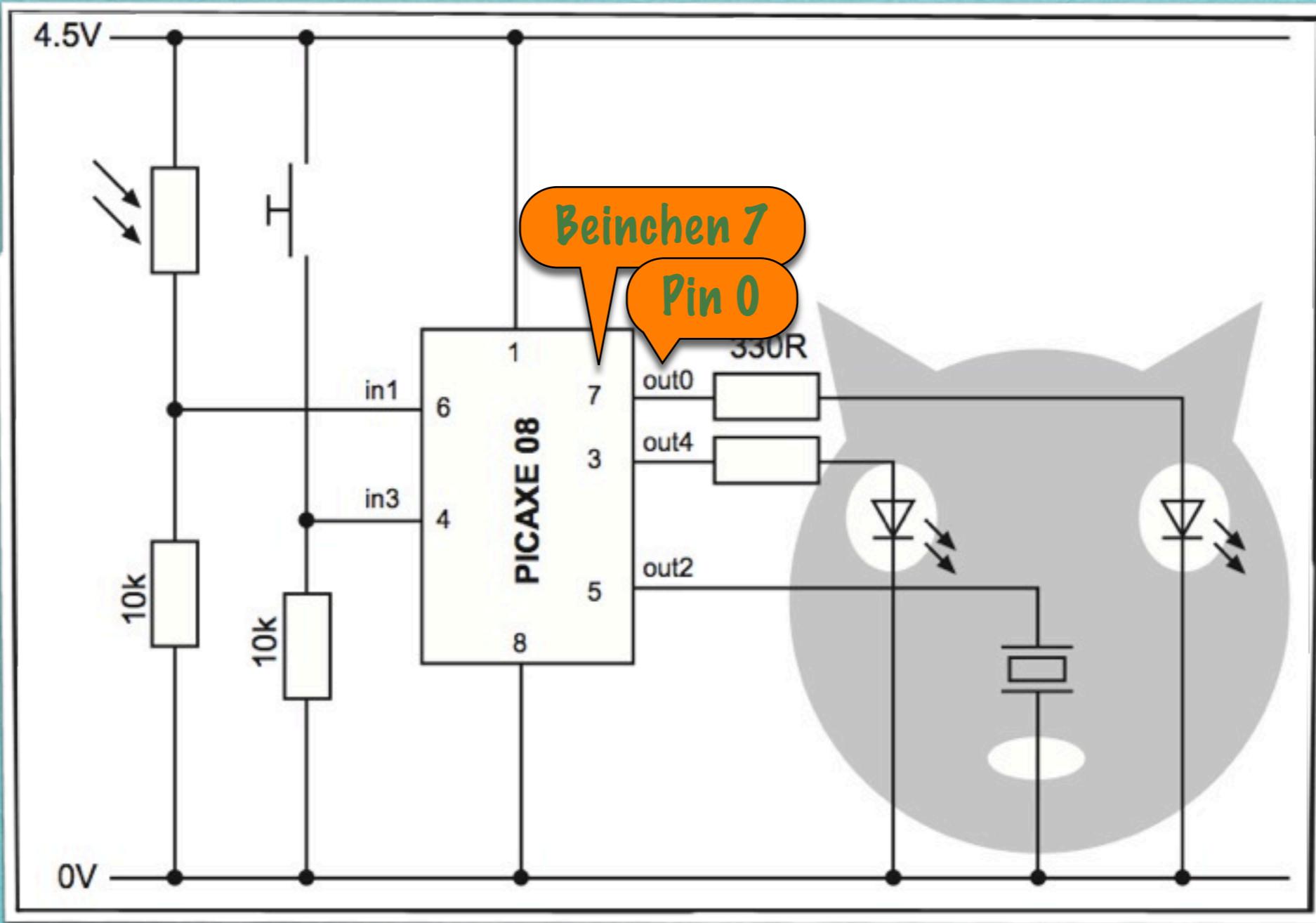


- ▶ Üben:
 - ⦿ Herunterladen von BASIC-Programmen
 - ⦿ Speichern eines Programms / Flußdiagramms
 - ⦿ Öffnen eines gespeicherten Programmes
 - ⦿ Ein neues BASIC-Programm erstellen
 - ⦿ Ein neues Flußdiagramm erstellen
 - ⦿ Bildschirm-Simulation eines Flußdiagrammes starten
 - ⦿ Ein Flußdiagramm in ein BASIC-Programm umwandeln
 - ⦿ Ausdrucken eines Programmes / Flußdiagrammes

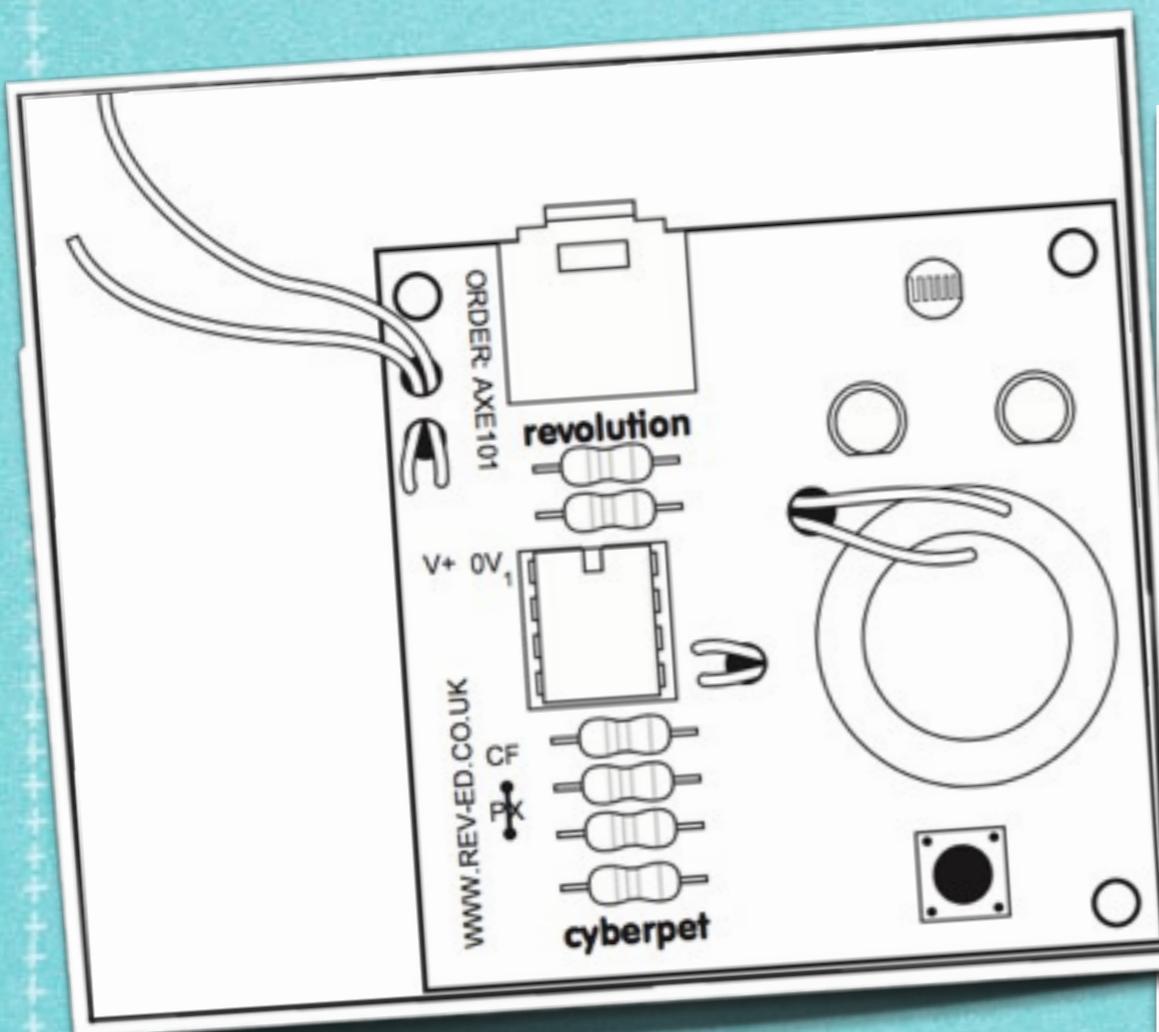
Blockdiagramm unseres Kuscheltiers



... und der Schaltplan



Aufbau der Schaltung



| Resistor colour codes | | Example shown: blue, grey, brown, gold = 680R ±5% | | |
|-----------------------|---|---|-----------------|-------------|
| Black | 0 | 0 | Black x1 | Silver ±10% |
| Brown | 1 | 1 | Brown x10 | Gold ±5% |
| Red | 2 | 2 | Red x100 | |
| Orange | 3 | 3 | Orange x1000 | |
| Yellow | 4 | 4 | Yellow x10,000 | |
| Green | 5 | 5 | Green x100,000 | |
| Blue | 6 | 6 | Blue x1,000,000 | |
| Violet | 7 | 7 | | |
| Grey | 8 | 8 | | |
| White | 9 | 9 | | |

Aufbau II

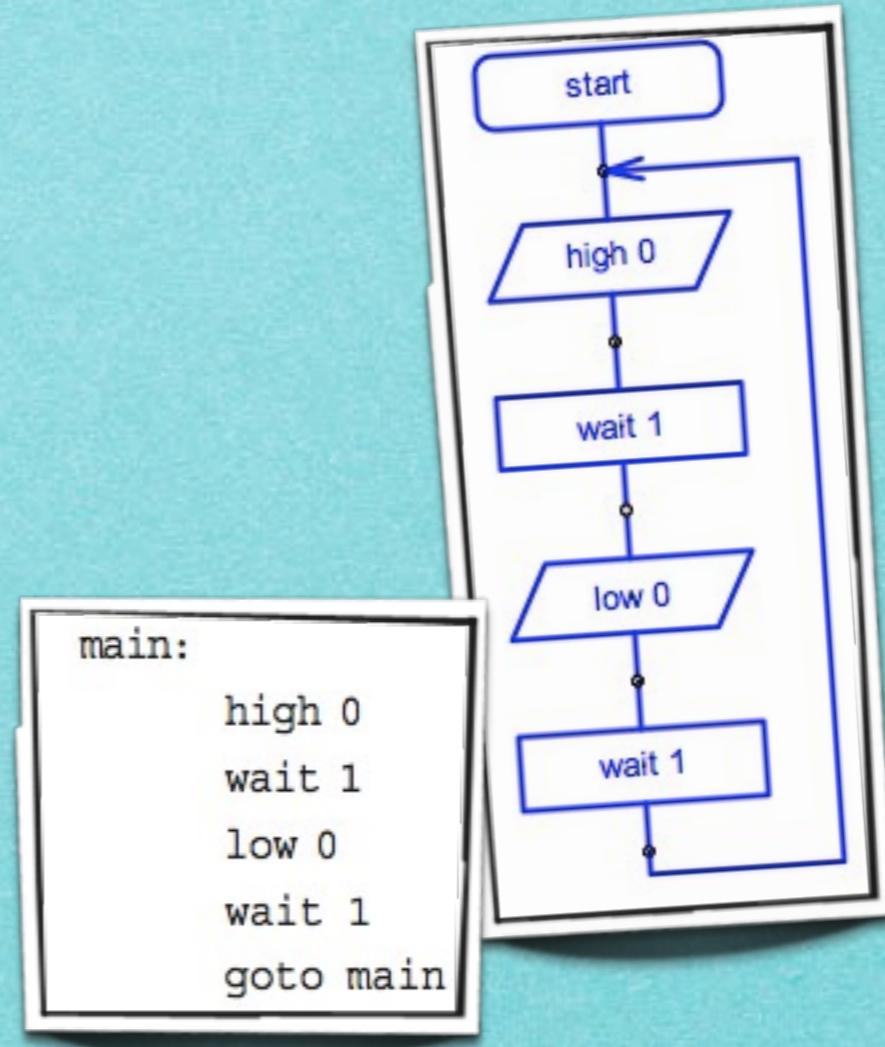


▶ Tipps

- ⦿ Mit den flachsten Bauteilen (die Widerstände) beginnen
- ⦿ Darauf achten, dass die Bauteile flach auf der Platine aufliegen.
Bei LEDs kann man die Beinchen nach aussen biegen
- ⦿ Die Stereo-Buchse muss auf der Platine „einrasten“
- ⦿ Achtet auf die Polung der LEDs
- ⦿ Nicht die Anschlußdrähte des Piezos überhitzen
- ⦿ Fädelt die Anschlüsse der Batterien durch die großen Löcher
- ⦿ Vergesst die Drahtbrücke nicht

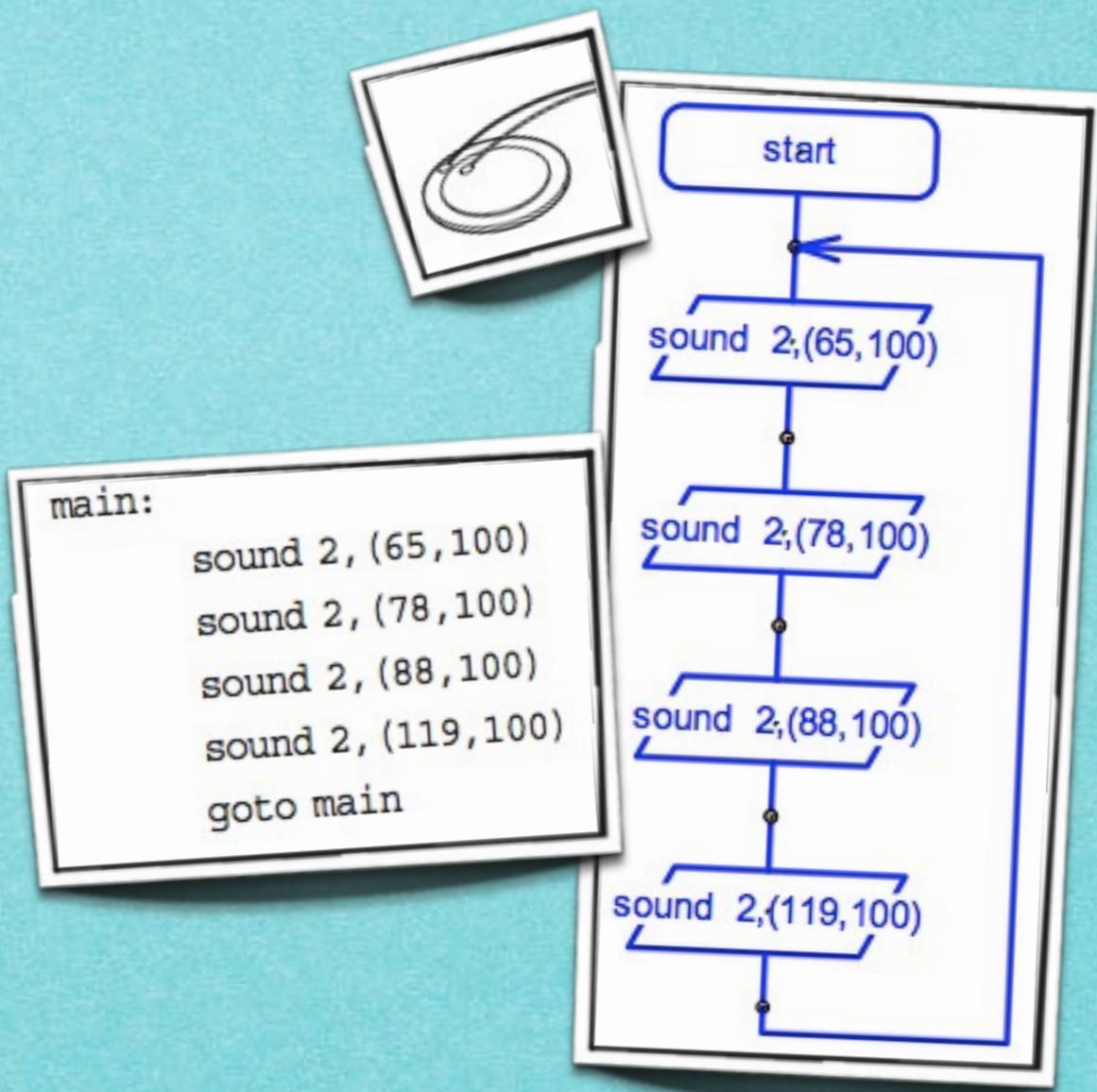
Programmieren: Test der LEDs 0 und 1

- ▶ Test LED 0
 - ⦿ Verbinde das USB-Kabel mit Computer und Platine
 - ⦿ Schalte die Programmierumgebung auf 08M-Modus und wähle den richtigen Port
 - ⦿ Tippe das nebenstehende Programm ein und lade es auf den PICAXE hoch
 - ⦿ Die LED 0 flickert beim Hochladen und blinkt dann jede Sekunde
- ▶ Wiederhole diesen Test, aber nutze *high 4* und *low 4* für die andere LED



Programmieren: Test des Piezo

- ▶ Tippe nebenstehendes Programm ein und lades es auf den PICAXE hoch
- ▶ Der Piezo sollte vier unterschiedliche Töne von sich geben



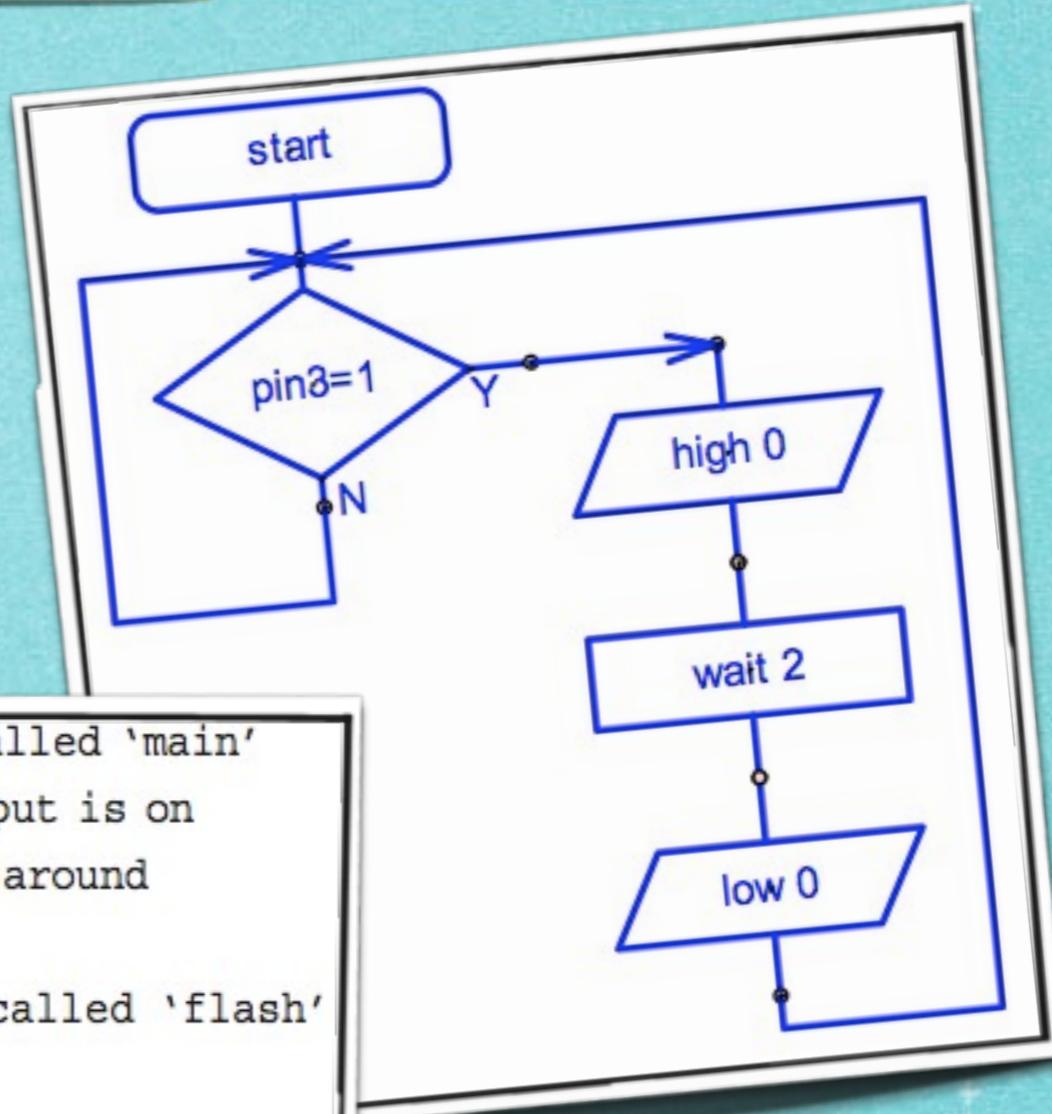
Programmieren: Test des Tasters

- ▶ Tippe das untenstehende Programm ein und lade es hoch
- ▶ Die erste LED sollte aufleuchten, wenn der Taster gedrückt wird

```
main:  
    if input3 is on then flash  
    goto main  
  
flash:  
    high 0  
    wait 2  
    low 0  
    goto main
```

' make a label called 'main'
' jump if the input is on
' else loop back around

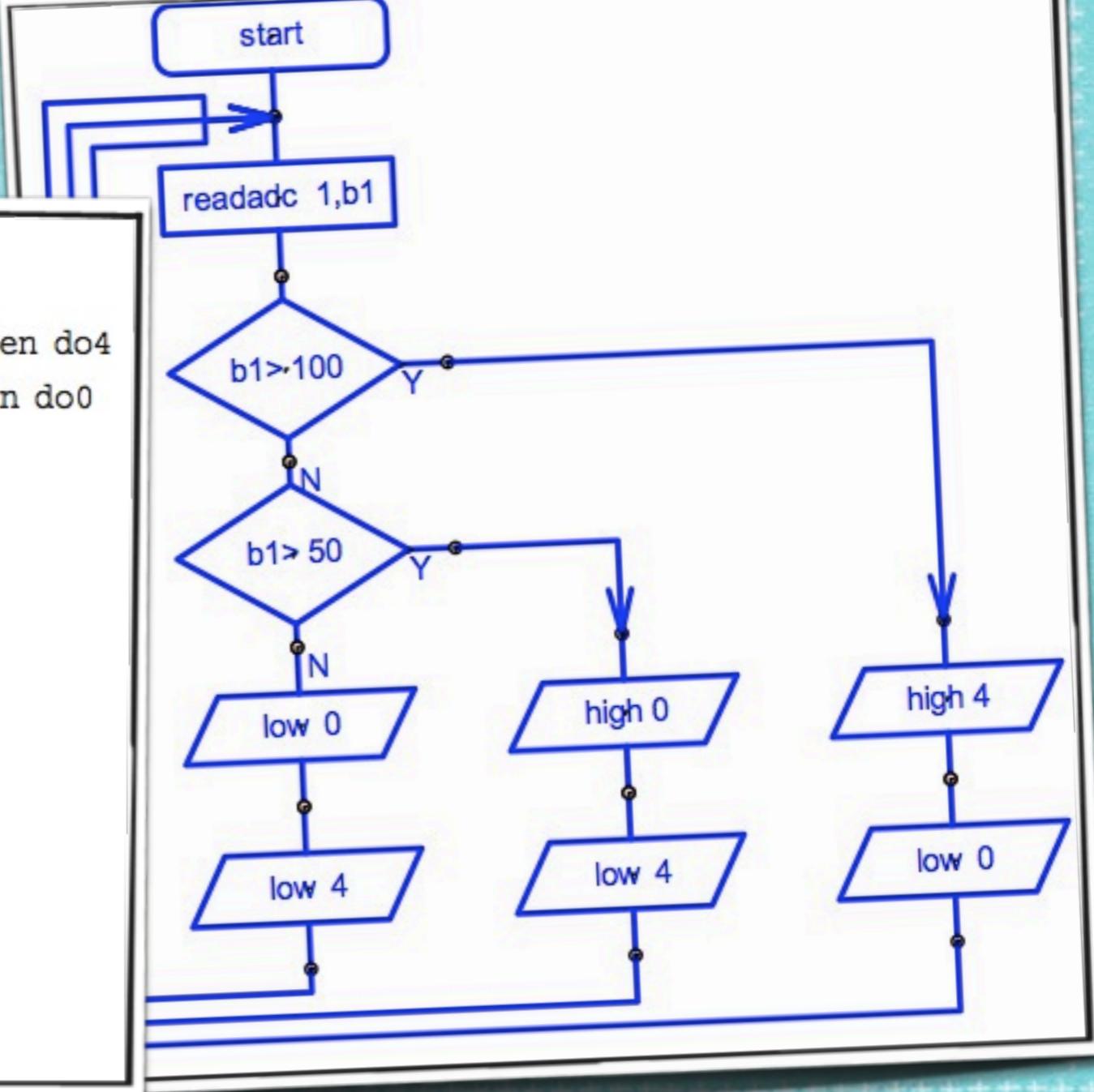
' make a label called 'flash'
' switch output 0 on
' wait 2 seconds
' switch output 0 off
' jump back to start



Programmieren: Test des LDR

- ▶ Tippe das nebenstehende Programm ein und lade es Hoch
- ▶ Evtl. musst du die Schwellwerte ändern. Versuche 60 und 30.
- ▶ Die LEDs sollen je nach Helligkeit in verschiedenen Mustern leuchten

```
main:  
    readadc 1,b1  
    if b1 > 100 then do4  
    if b1 > 50 then do0  
    low 0  
    low 4  
    goto main  
  
do4:  
    high 4  
    low 0  
    goto main  
  
do0:  
    high 0  
    low 4  
    goto main
```



Testergebnis

- ▶ Alles funktioniert!
- ▶ Jetzt sei kreativ und gibt
deinem Kuscheltier eine
Persönlichkeit.

Hier zwei Beispiele ...



Programmier-Ideen I

Program 1

```
' ***** main loop *****  
' loop here flashing lights  
' and checking switch and light sensor  
  
main:  
' LEDs full on and read light value  
    high 4  
    high 0  
    readadc 1,b1  
  
' if light value low then go to bed  
    if b1 < 40 then bed  
  
' if switch pushed do sound  
    if pin3 = 1 then purr  
  
' do a delay  
    pause 500  
  
' LEDs full off and check sensor again  
    low 4  
    low 0  
    readadc 1,b1
```

```
' if light value low then go to bed  
    if b1 < 40 then bed  
  
' if switch pushed do sound  
    if pin3 = 1 then purr  
    pause 500  
    goto main  
  
' ***** make sound *****  
purr:  
    sound 2, (120,50,80,50,120,50)  
    pause 200  
    goto main  
  
' ***** bed routine when in dark *****  
' in dark so switch off LEDs and wait till  
' light again  
  
bed:  
    low 0  
    low 4  
    readadc 1,b1  
    if b1 > 40 then main  
    goto bed
```

Programmier-Ideen I

- ▶ Das Programm I blinkt die LED -Augen in einer Schleife und prüft dabei den Taster
- ▶ Bei gedrücktem Taster wird ein Ton erzeugt. Du musst den Schalter gedrückt halten, bis der Ton erklingt
- ▶ Bei abgedecktem LDR schläft das Kuscheltier ein

Programmier-Ideen II

- Das Programm II ist viel fortschrittlicher. In seiner Hauptschleife blendet er die LEDs weich ein und aus und prüft gleichzeitig Taster und LED
-
- Bei gedrücktem Taster wird ein Ton erzeugt. Du musst den Schalter gedrückt halten, bis der Ton erklingt
- Bei gedrücktem Taster erfolgt eine Klanausgabe mit `sound`-Befehl
- Nach drei Tastendrücken wird eine Melodie gespielt
- Bei abgedecktem LDR schläft das Kuscheltier ein

Werde selber kreativ!

- ▶ Ändere das Verhalten deines Kuscheltieres
- ▶ Bastele ihm ein witziges Äußeres



Programmier-Ideen II

Program 2

```
' ***** main loop *****  
  
main:  
  
' first make LED 0 brighter in 10 steps  
    for b2 = 1 to 10  
        pwm 0, b2, 20  
    next b2  
    high 0  
  
' then make LED 4 brighter in 10 steps  
    for b2 = 1 to 10  
        pwm 4, b2, 20  
    next b2  
    high 4  
  
' LEDs full on so read light value  
    readadc 1,b1  
  
' if light value low then goto bed  
    if b1 < 40 then bed  
  
' if switch pushed add petting value  
    if pin3 = 1 then addpet  
    goto main  
  
' ***** end of main loop *****
```

Weitere Informationen



- ▶ Weitere Informationen zum Workshop am 3./4.6.2010 unter <http://tinkerthon.de/gruppe/bonn>
- ▶ Anmeldung unter <http://www.muellestumpe.de/>
- ▶ Dieser Workshop basiert auf dem „Cyberpet Project“
<http://www.rev-ed.co.uk/docs/axe101.pdf>
- ▶ Das ausführliche PICAXE-Handbuch
http://www.rev-ed.co.uk/docs/picaxe_manual2.pdf