

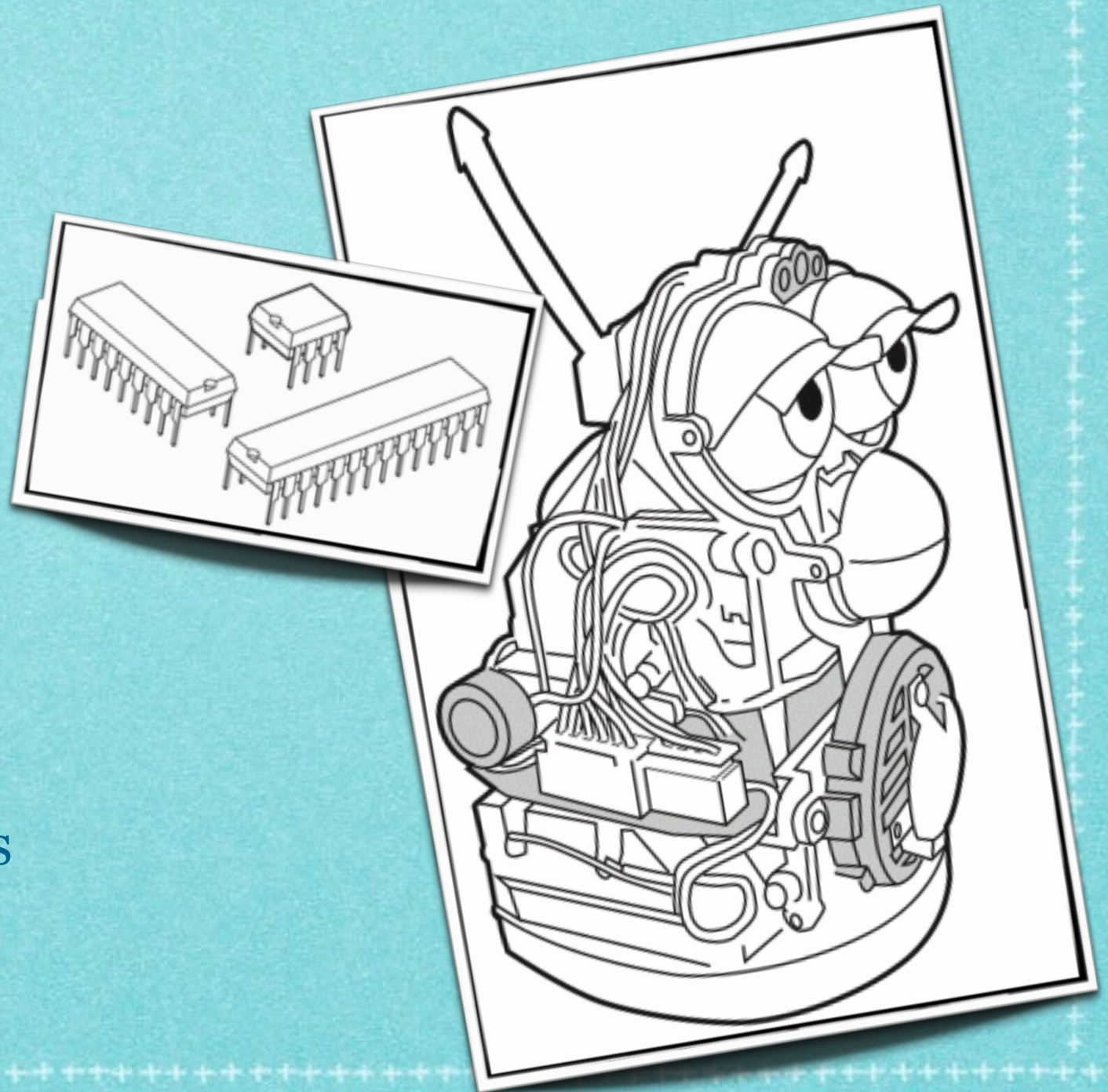
- Der heiße Draht - Physical Computing

Workshop Oktober 2014

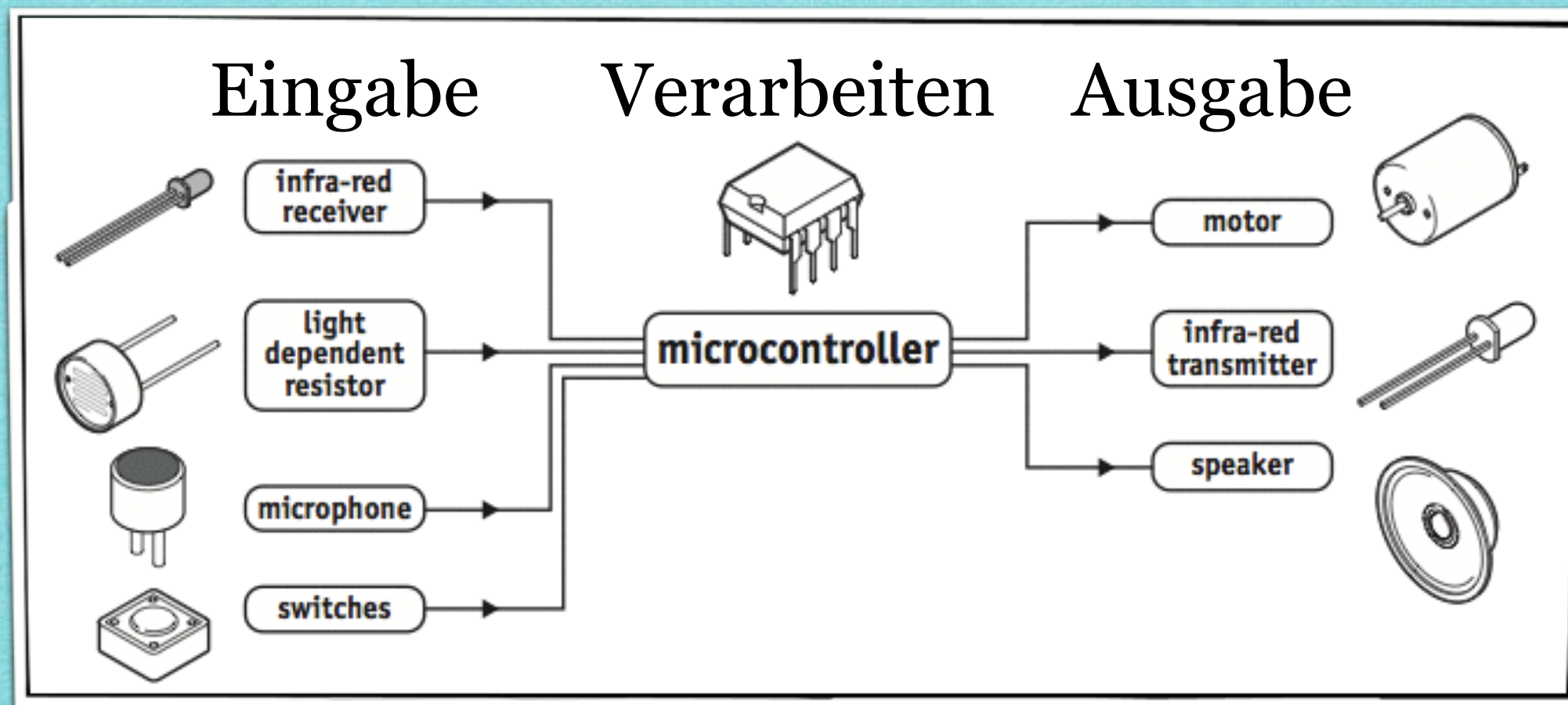
Olav Schettler <olav@tinkerthon.de>

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



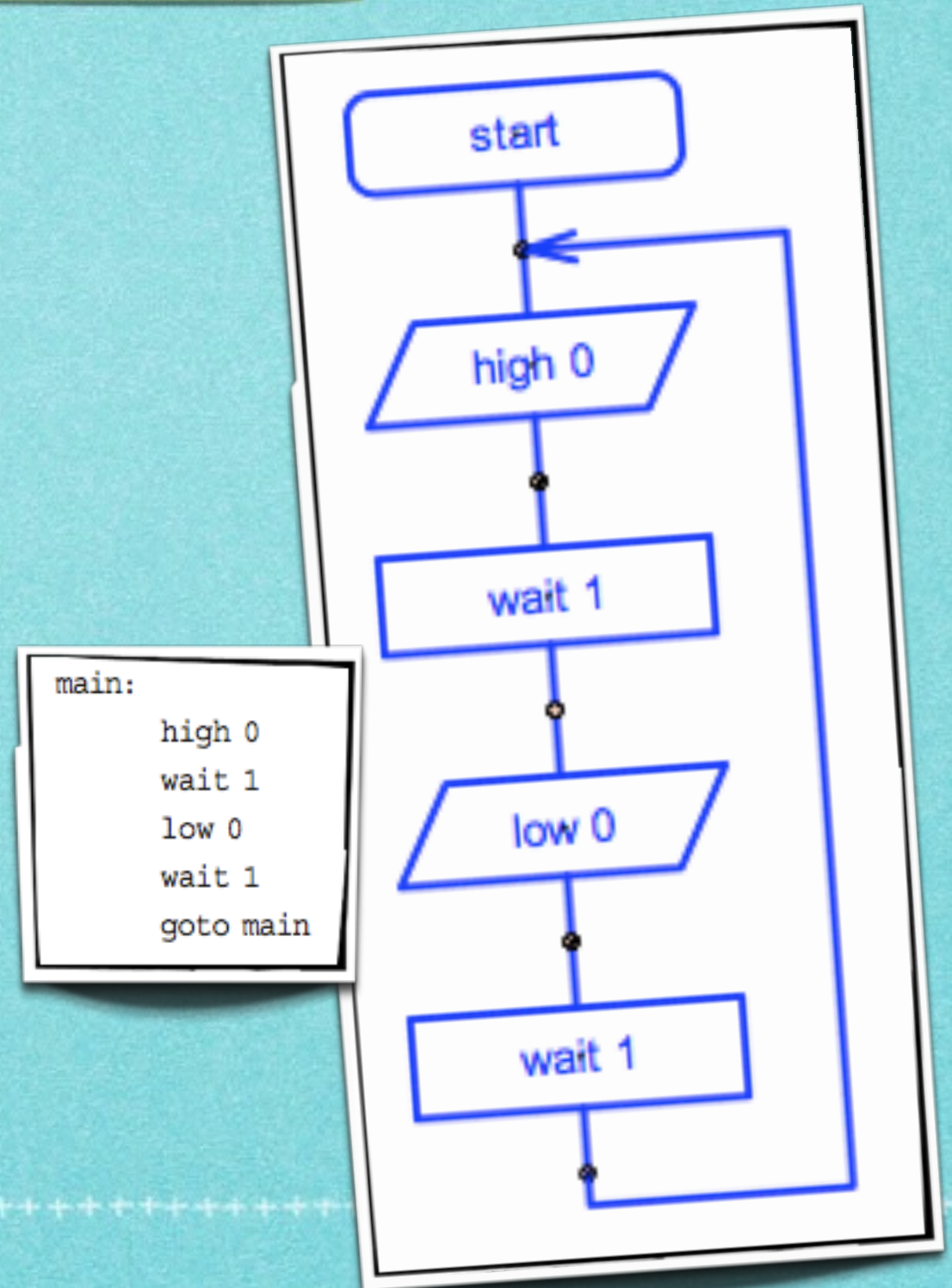
Blockdiagramme



- 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

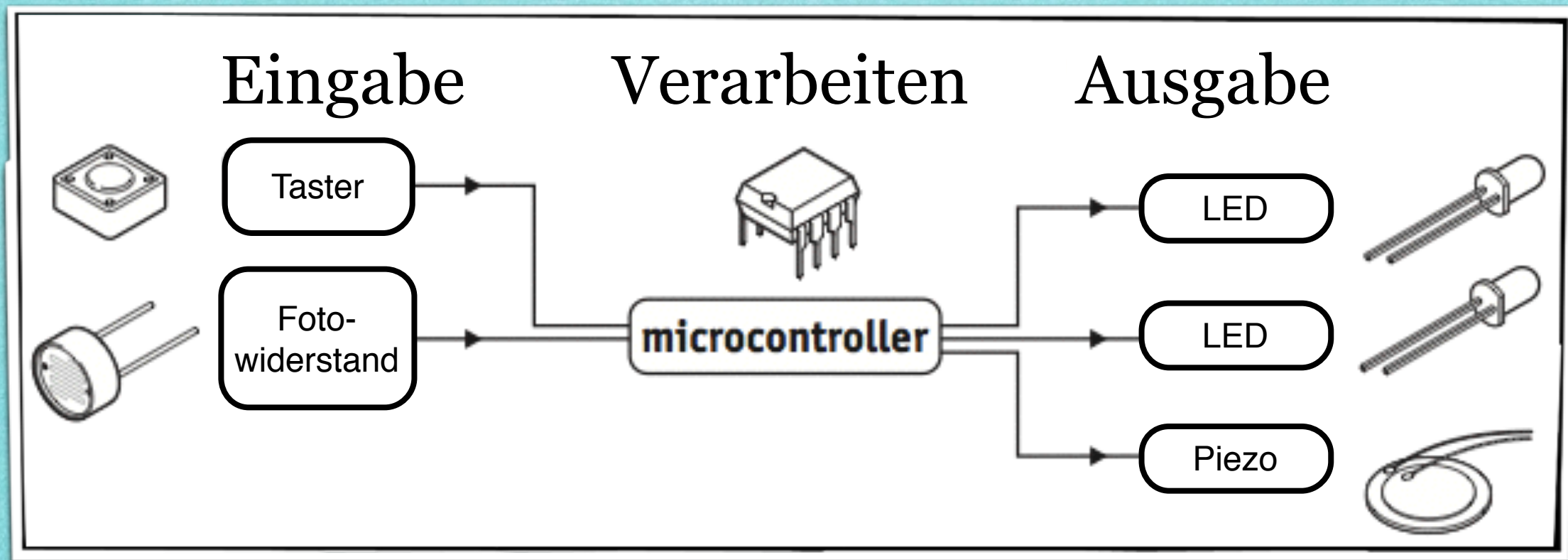


Wir bauen ein elektronisches Spiel

- ▶ Gehirn: ein PICAXE-08M2
- ▶ LEDs zur Anzeige von Zeit und Berührungen, Piezo-Wandler für Töne
- ▶ Reagiert auf Berührung und Zeit



Blockdiagramm unseres Spiels



Elektronische Bauteile

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

PICAXE-08 or 08M microcontroller



light emitting diode (LED)



piezo sounder



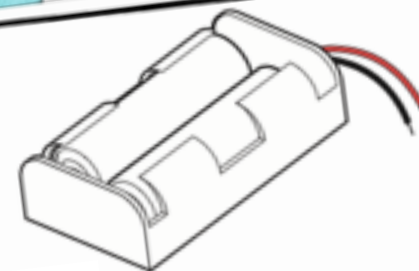
light dependent resistor (LDR)



switch



batteries



Wir brauchen ausserdem:

picaxe download socket

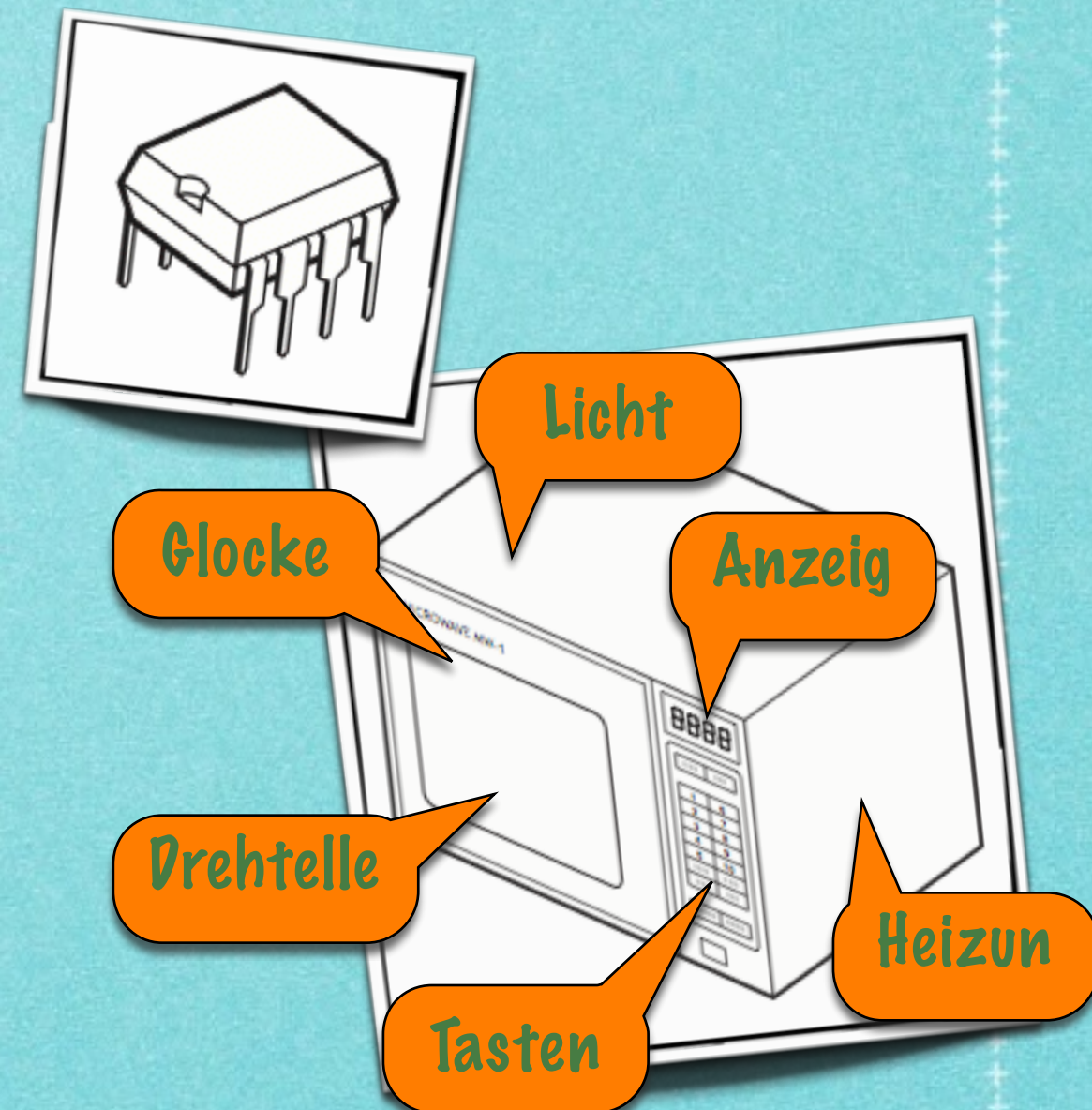


resistors

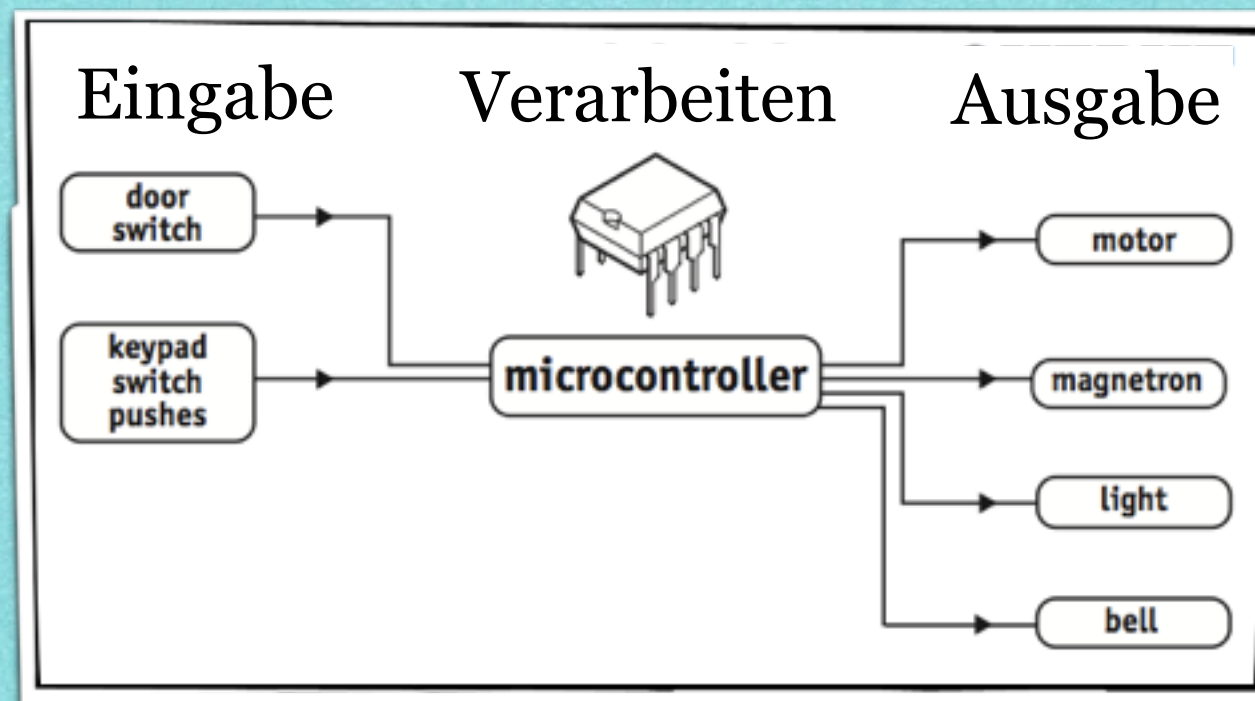


Microcontroller

- ▶ 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

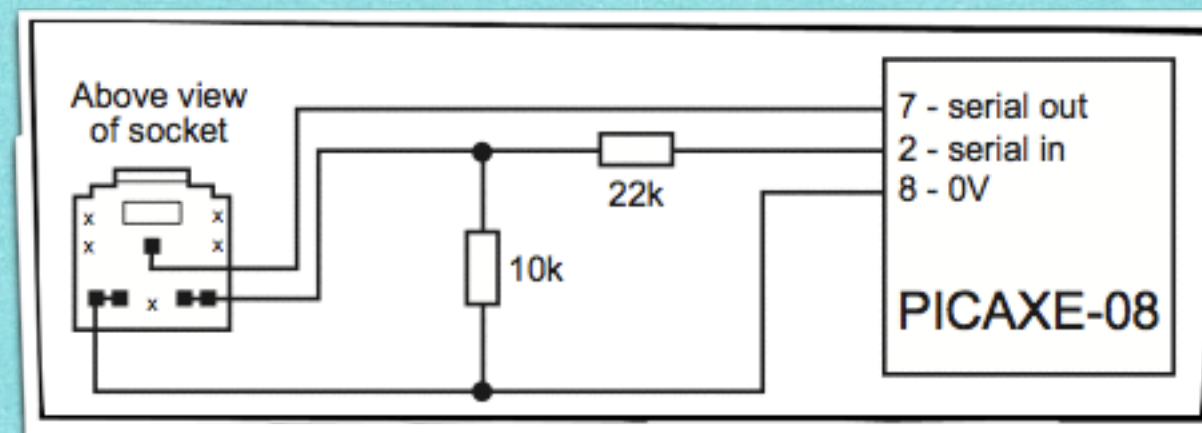


Wie kommen Programme auf den Microcontroller?



- ▶ 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

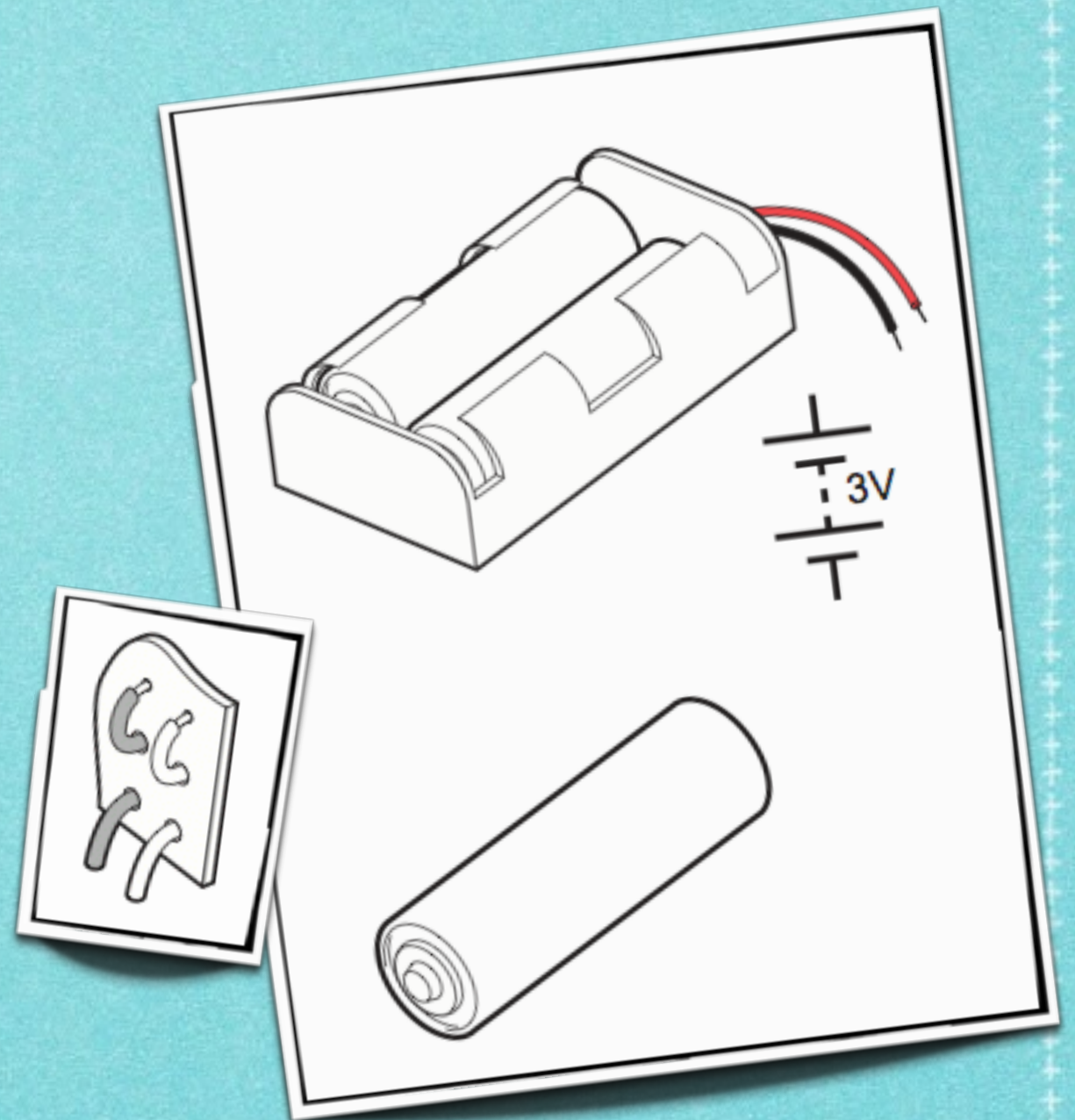
Wie werden Programme hochgeladen?



- ▶ 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

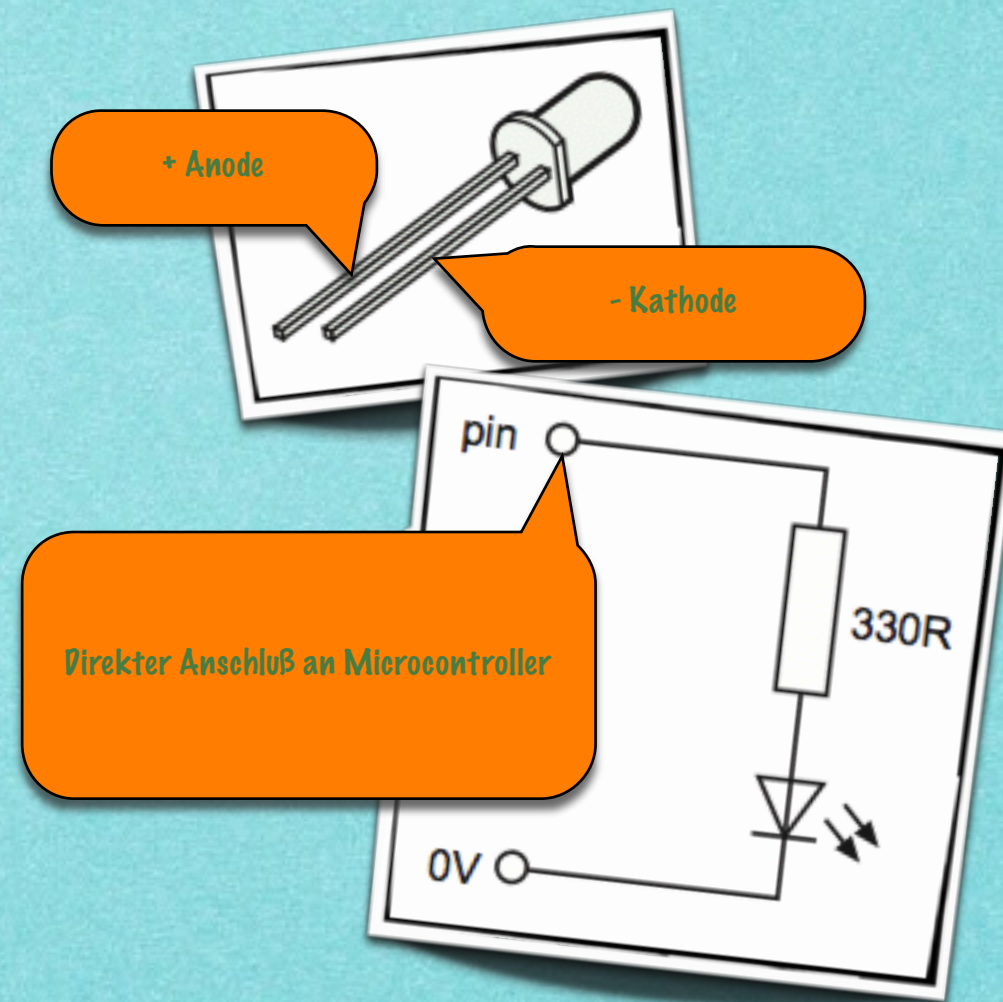
Batterien

- ▶ 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



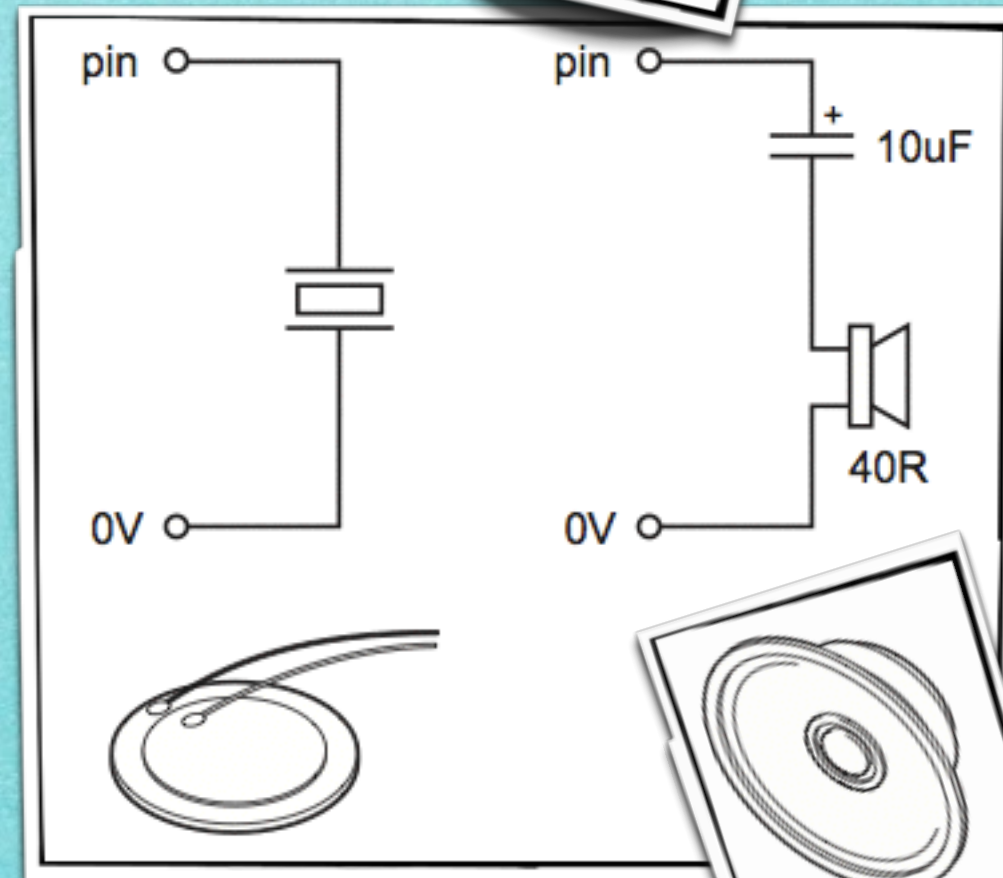
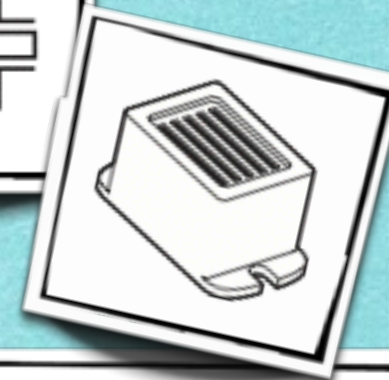
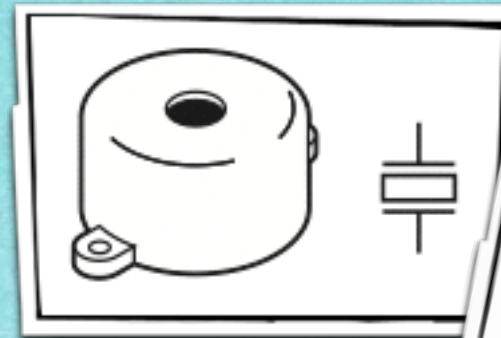
LEDs - Leuchtdioden

- ▶ 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

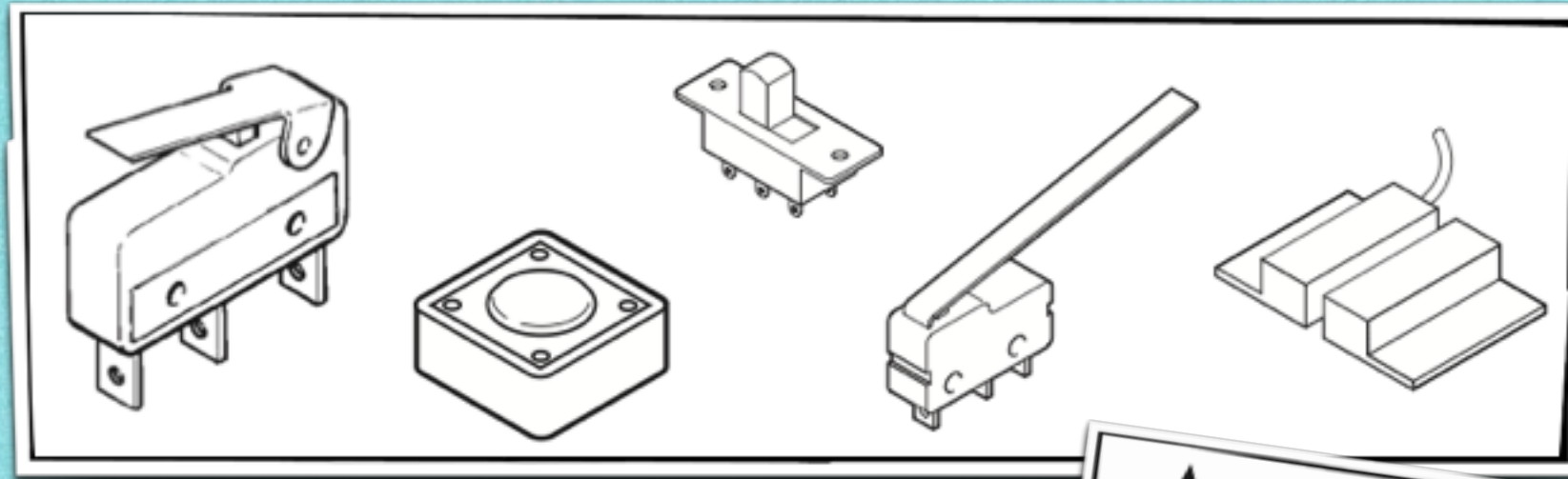


Piezo-Schallwandler

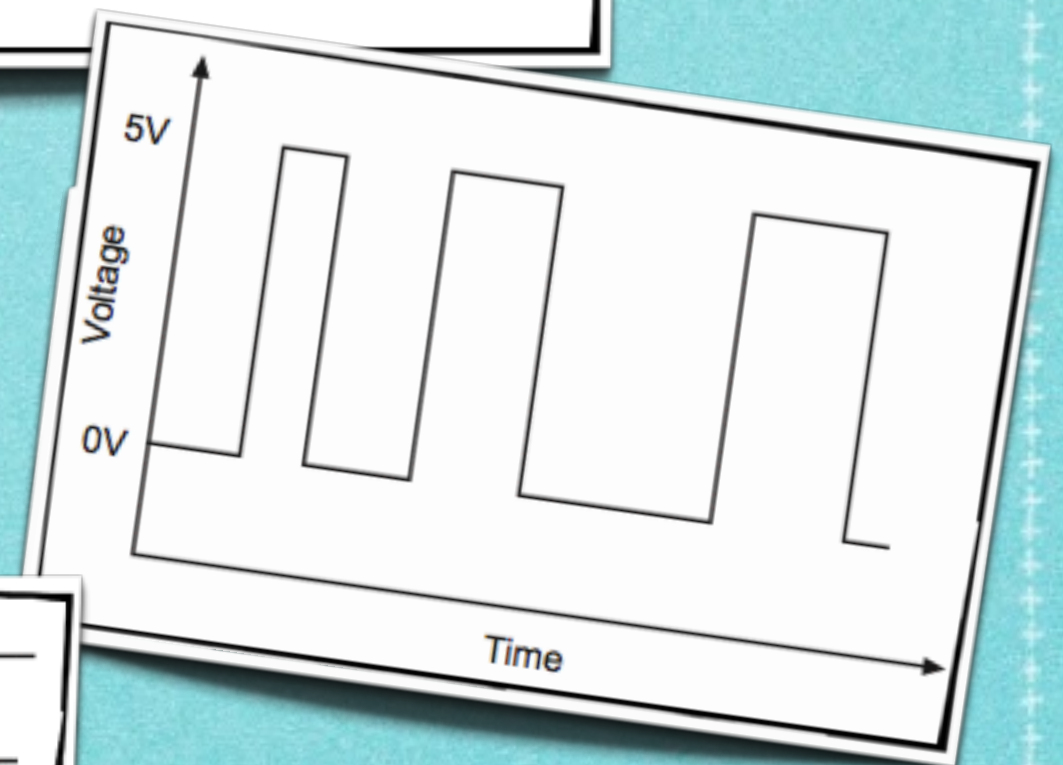
- ▶ Piezo-Schallwandler sind preiswerte Mini-Lautsprecher
- ▶ Einsatz zur Rückmeldung an Nutzer oder in klingenden Geburtstagskarten
- ▶ Piezos können direkt mit einem Microcontroller-Pin verbunden werden
- ▶ Für lautere Klänge kann ein Lautsprecher mit Kondensator benutzt werden



Digitale Fühler (Schalter)

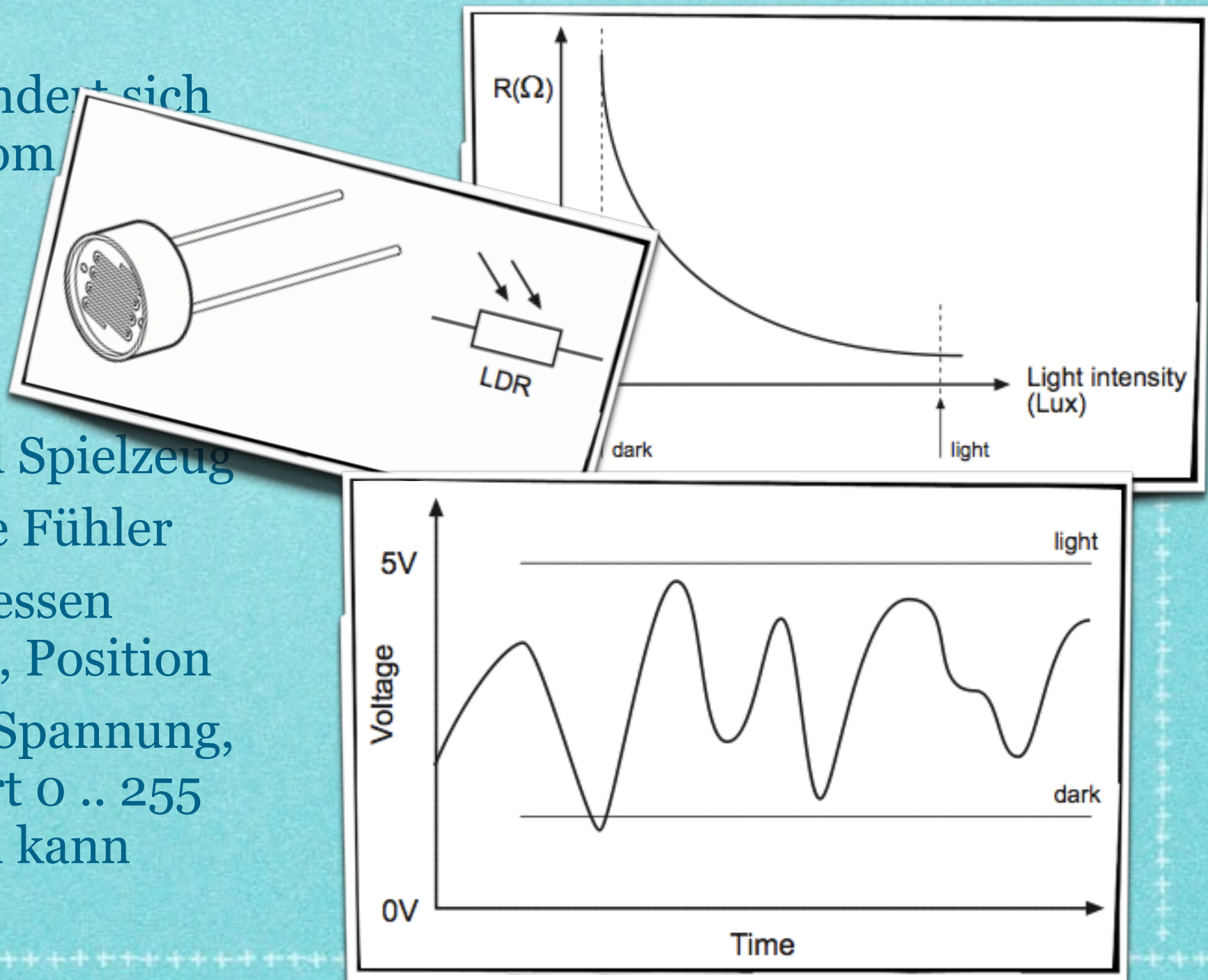


- ▶ Drucktaster, Microschalter, Reed-Schalter, Kippschalter
- ▶ Anwendung in Tastaturen, Alarmanlagen, Türschließung, Bewegungsfühler

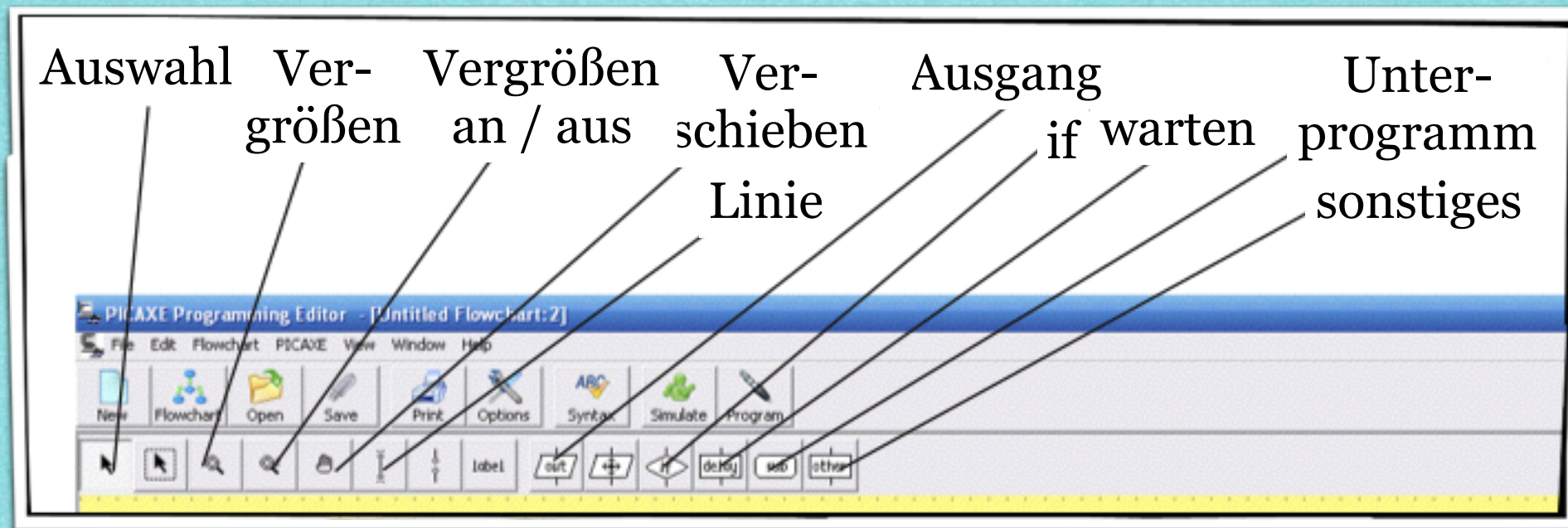


Lichtempfindliche Widerstände (LDR)

- ▶ 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



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 Programmmeditor 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

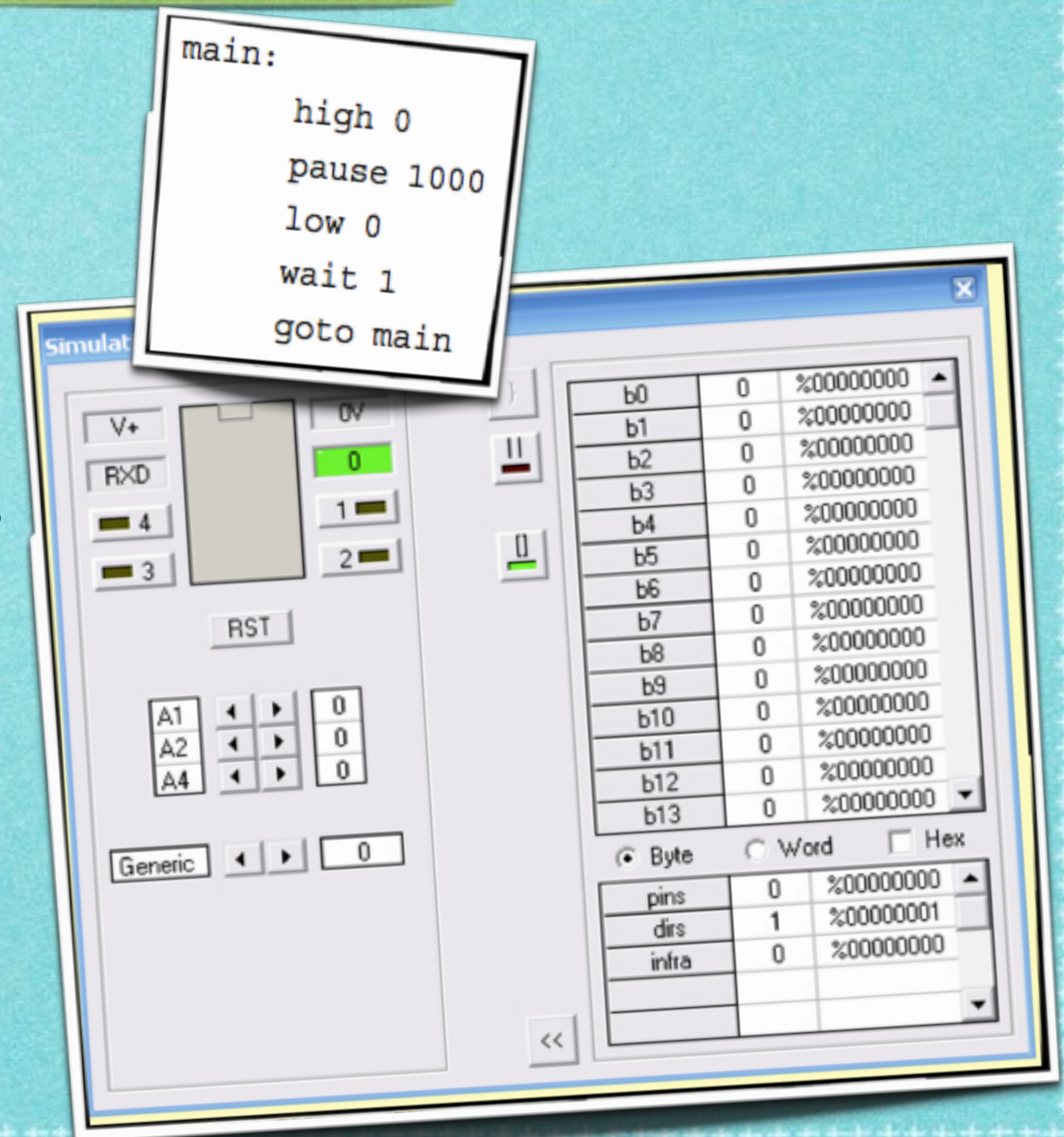


Flußdiagramme hochladen

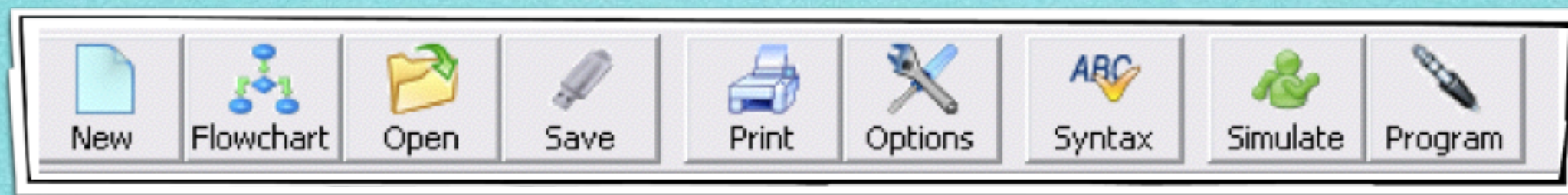
- ▶ Flußdiagramme können direkt auf den Microcontroller geladen werden:
 - ▶ Menüpunkt „Programm“
- ▶ Bitte beachten:
 - ▶ Das Wandeln nach BASIC bricht bei nicht verbundenen Symbole ab
 - ▶ Immer ein Stop-Symbol benutzen
 - ▶ Umwandeln / Herunterladen: 2x F5 drücken

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



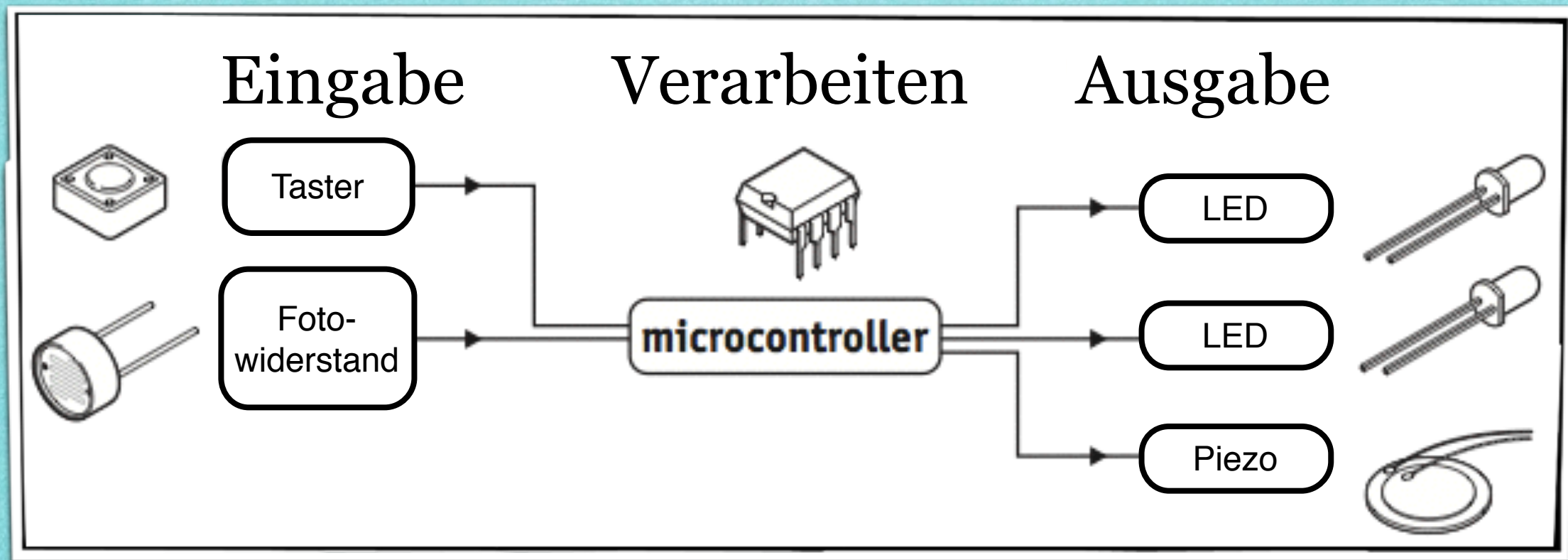
Programmierungsumgebung



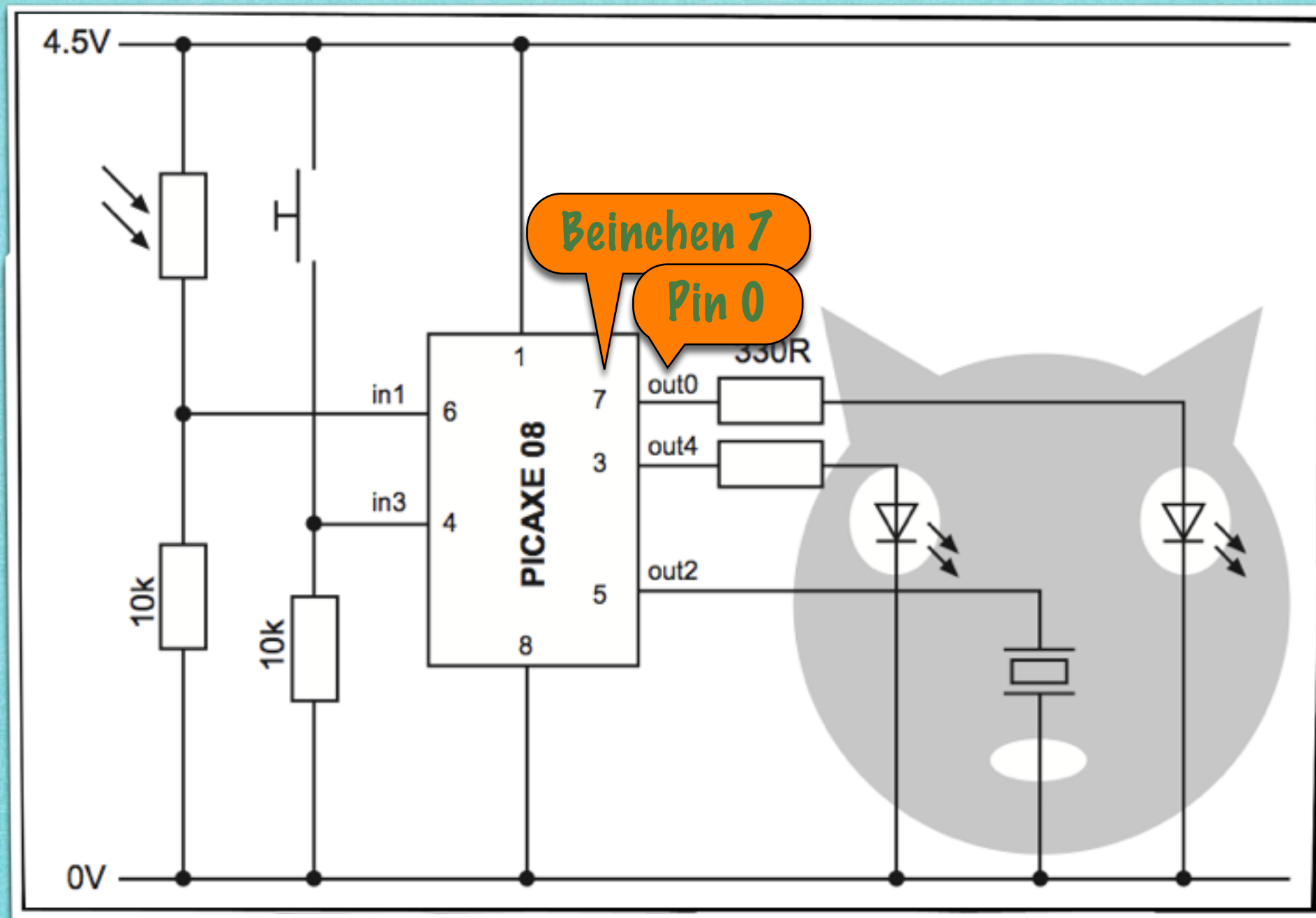
► Üben:

- 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 Spiels

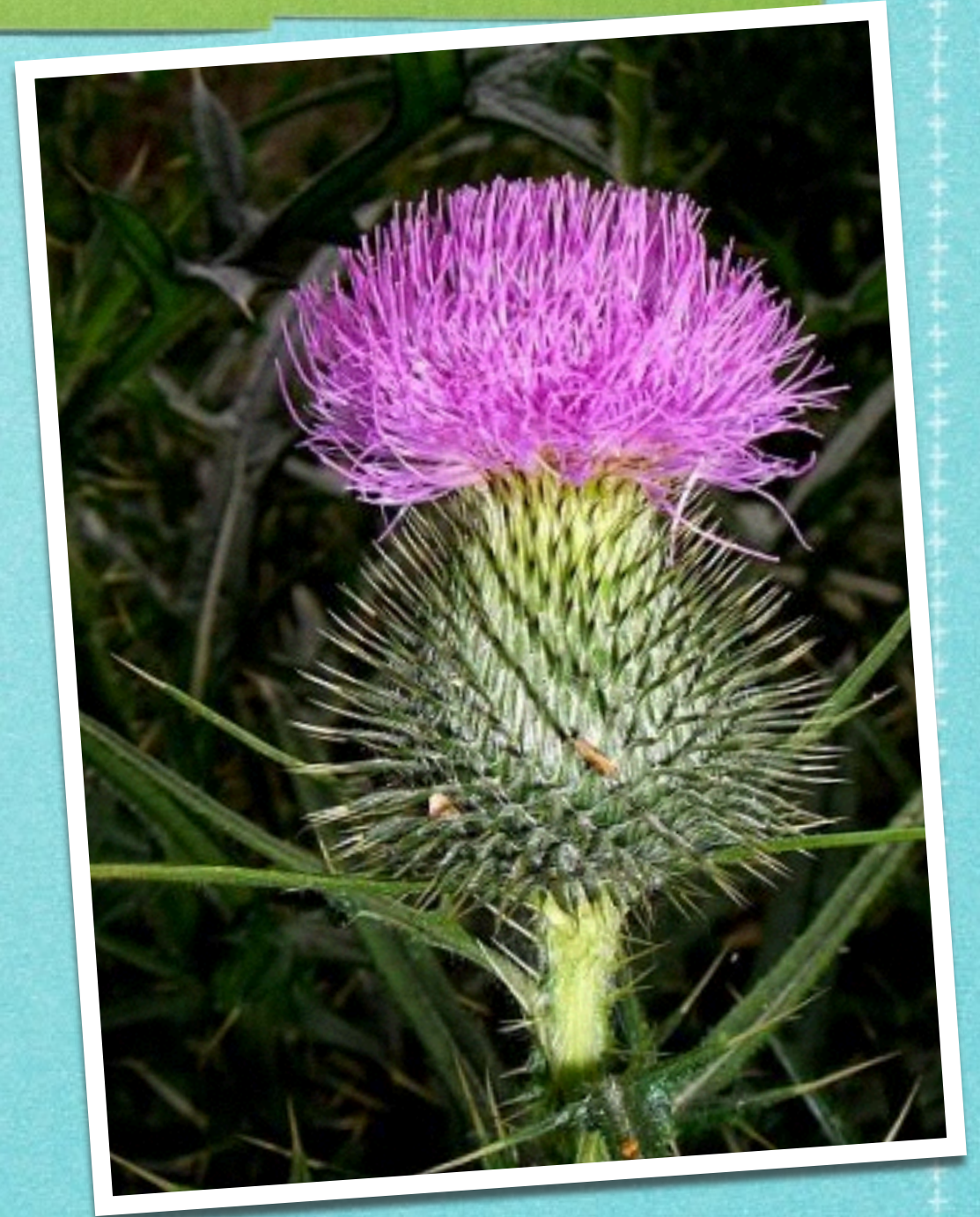


... und der Schaltplan



Testen der einzelnen Bausteine

- Um sicher zu sein, dass alles funktioniert, probieren wir jetzt die einzelnen Ein- und Ausgänge aus



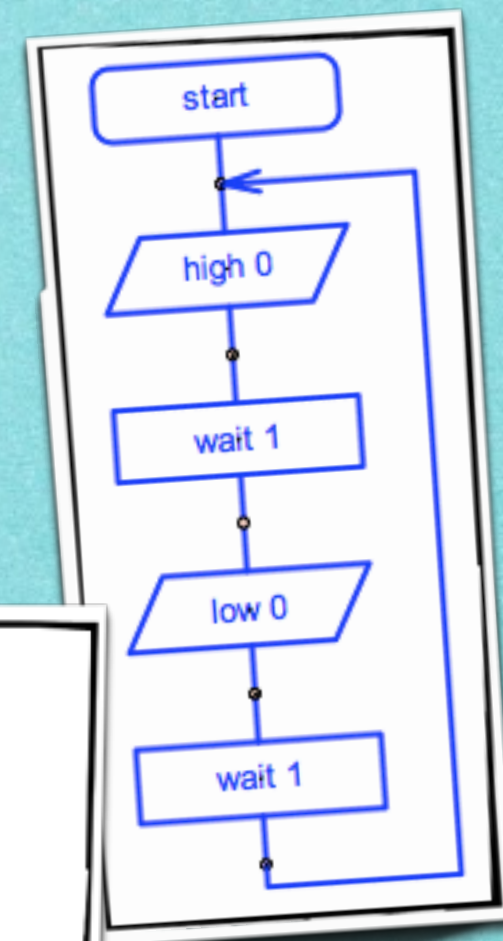
Programmieren: Test der LEDs 0 und 1

► 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

```
main:  
  high 0  
  wait 1  
  low 0  
  wait 1  
  goto main
```



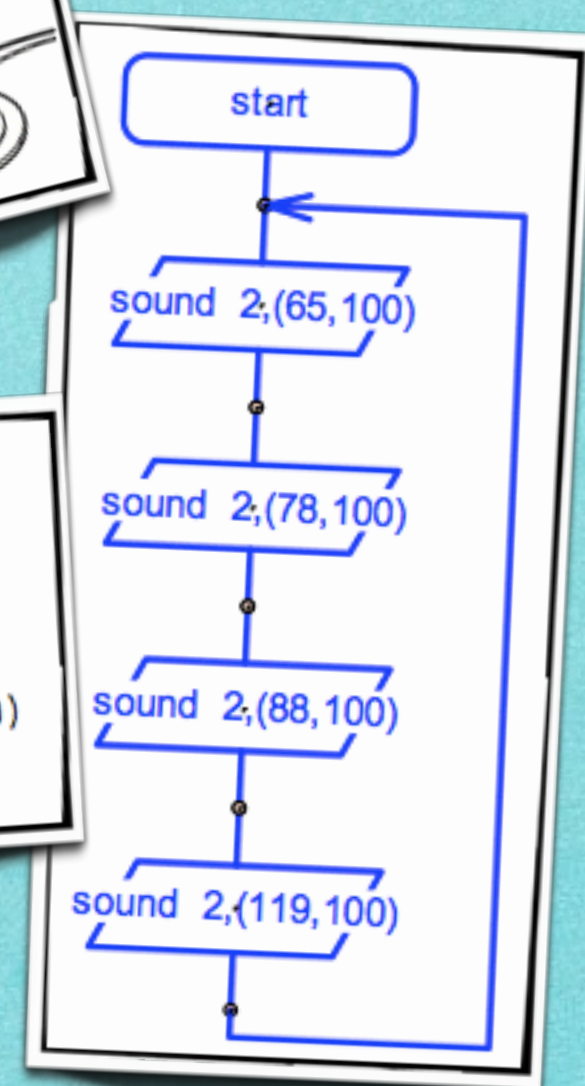
Programmieren: Test des Piezo

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



main:

```
sound 2, (65,100)  
sound 2, (78,100)  
sound 2, (88,100)  
sound 2, (119,100)  
goto main
```

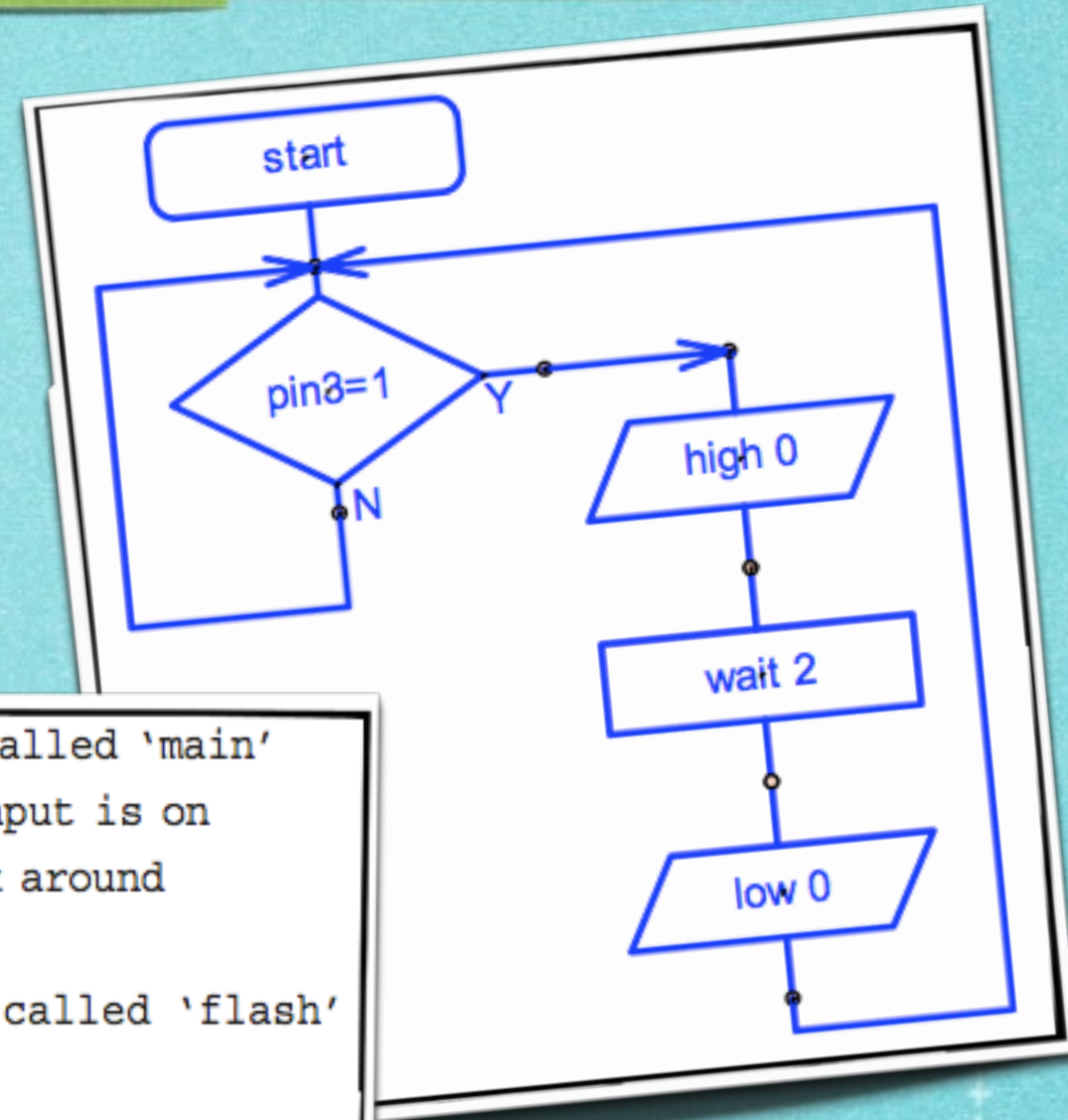


Programmieren: Test des Tasters

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

```
main:                                ' make a label called 'main'
    if input3 is on then flash        ' jump if the input is on
    goto main                        ' else loop back around

flash:                                ' make a label called 'flash'
    high 0                            ' switch output 0 on
    wait 2                            ' wait 2 seconds
    low 0                             ' switch output 0 off
    goto main                         ' jump back to start
```



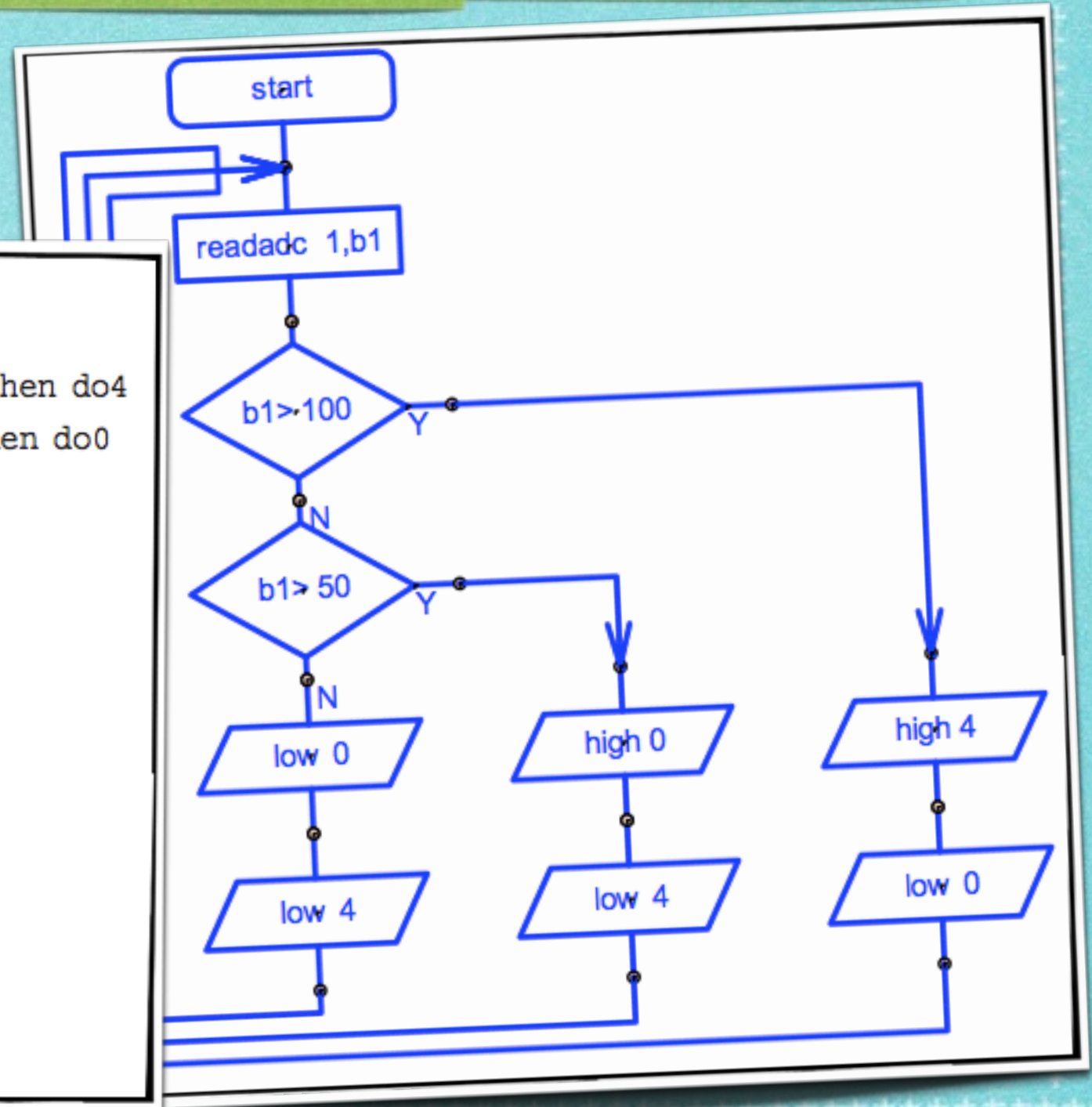
Programmieren: Test des LDR

- ▶ 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 gehen wir daran, unser Spiel zu programmieren ...



Erste Version

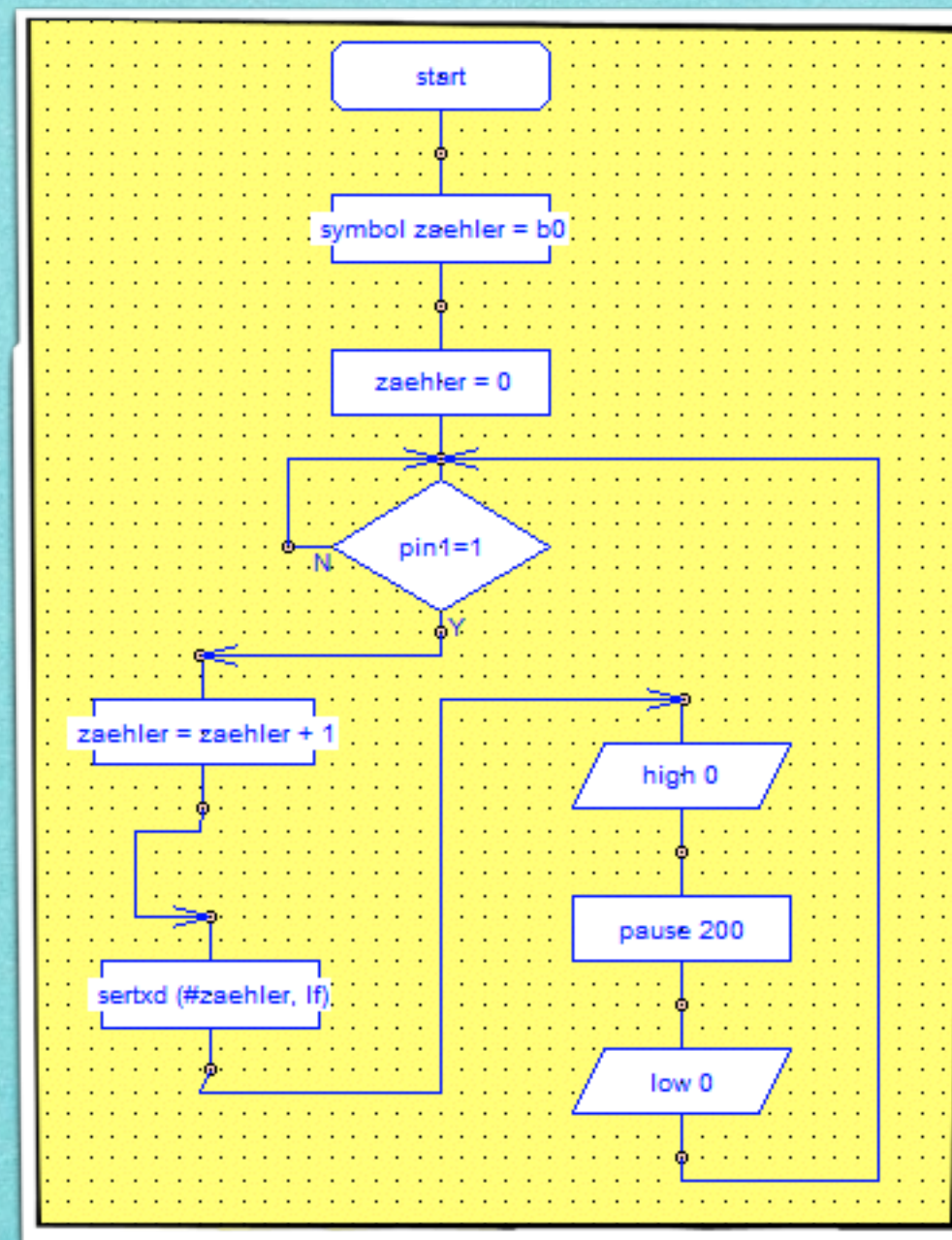
```
; hd01
; Heisser Draht 01
do
    if pin1 is on then
        high 0
    else
        low 0
    endif
loop
```


Zweite Version

```
; hd02
; Heisser Draht 02
symbol zaehler = b0      ' Variablenname zur besseren Lesbarkeit
zaehler = 0

do
  if pin1 is on then
    ' Dauert Berührung länger als 0,2s,
    ' wird sie mehrfach gezählt
    zaehler = zaehler + 1
    ' Ausgabe des Zählers
    srtxd (#zaehler, lf)
    high 0
    pause 200  ' LED blinkt kurz
    low 0
  endif
loop
```


Zweite Version als Flußdiagramm



3. Version: Punktezähler

```
; hd03
; Heisser Draht 04
symbol zaehler = b0 ' Variablenname zur besseren Lesbarkeit
symbol zeit = b1
symbol punkte = w2
zaehler = 0
zeit = 0

do
    zeit = zeit + 1
    if pin1 is on then
        zaehler = zaehler + 1
    high 0
    endif
    pause 200 ' LED blinkt kurz
    low 0
loop until pin3 is on

' punkte = 100 - (zaehler * 10) - zeit
punkte = zaehler * 10
punkte = 1000 - punkte
punkte = punkte - zeit
sertxd ("Punkte: ", #punkte, ", Zaehler: ",
#zaehler, ", Zeit; ", #zeit, lf)
```


3. Version: Punktezähler

```
; hd03
; Heisser Draht 04
symbol zaehler = b0 ' Variablenname zur besseren Lesbarkeit
symbol zeit = b1
symbol punkte = w2
zaehler = 0
zeit = 0

do
    zeit = zeit + 1
    if pin1 is on then
        zaehler = zaehler + 1
    high 0
    endif
    pause 200 ' LED blinkt kurz
    low 0
loop until pin3 is on

' punkte = 100 - (zaehler * 10) - zeit
punkte = zaehler * 10
punkte = 1000 - punkte
punkte = punkte - zeit
sertxd ("Punkte: ", #punkte, ", Zaehler: ",
#zaehler, ", Zeit; ", #zeit, lf)
```


Komplett!

- ▶ s. Texteditor ...
- ▶ Zwei Programme gleichzeitig
- ▶ Zeitmessung
- ▶ Berührungen zählen
- ▶ Punkte ausgeben



Weitere Informationen



- ▶ Weitere Informationen zum Workshop am 08./09.10.2014 unter <http://tinkerthon.de/>
- ▶ Die Hardware basiert auf dem „Cyberpet Project“ <http://www.picaxe.com/Hardware/Project-Kits/Cyberpet-Project-Kit/>
- ▶ Kostenlose Programmierumgebung und Handbuch <http://www.picaxe.com/>
- ▶ Code-Beispiele: <https://github.com/tinkerthon/Der-heisse-Draht>