

## ARBEITSBLATT

### KOMPONENTENTEST

Alle Komponenten der Leuchtscheibe-mini testen

### VORAUSSETZUNG

---

Du hast die Leuchtscheibe-mini fertig zusammengebaut.

### AUFGABE

---

Teste ob alle Komponenten der Leuchtscheibe-mini richtig funktionieren.

# Test

- CodeBug
- Taste A
- LED-Ring
- rote Taste

Gehe auch in dieser Aufgabe schrittweise vor.

- Teste die 5x5 roten LED vom CodeBug. Zünde jede der roten LED an und lösche sie wieder ab. Schalte die LED mit den beiden Tasten A und B an und ab.

- Teste die vier farbigen Tasten und den LED-Ring. Fülle den LED-Ring in der Farbe der gedrückten Taste langsam auf.

### TIPPS

---

Der Komponententest ist die letzte Kontrolle beim Bau der Leuchtscheibe-mini.

Wenn dies das erste Programm ist, welches du selbst schreibst, dann schau die fertige Musterlösung an. Wo im Programm werden welche Komponente getestet?

## BLOCKLISTE

Liste der wichtigsten Blöcke

---

Start-Block



Einstellungen für den CodeBug machen:

- Anschlüsse als digitale Eingänge einstellen
- Pullup für die drei externen Tasten 0, 2, 3  
Die Taste 1 braucht einen eigenen externen Widerstand.
- LED-Ring über «ColourStar» aktivieren

# ARBEITSBLATT

## KOMPONENTENTEST

### Musterlösung (Teil 1 von 2)

The Scratch script is as follows:

- Start** block with settings: Richtung **hoch**, schlafe **niemals**, Beinchen 0 auf **Digitaleingang**, Beinchen 1 auf **Digitaleingang**, Beinchen 2 auf **Digitaleingang**, Beinchen 3 auf **Digitaleingang**, Beinchen 0 Pullup **aktiv**, Beinchen 2 Pullup **aktiv**, Beinchen 3 Pullup **aktiv**, aktiv **ColourStar (als Schweif)**.
- Four **Schreibe** blocks: **TasteGedrückt** (falsch), **FarbanteilRot** (0), **FarbanteilGruen** (0), **FarbanteilBlau** (0).
- bewege Motiv** block: **erzeuge Text-Motiv** (Komponententest), **Schreibrichtung** (rechts →), **Verzögerung (ms)** (150), **Bewegungsrichtung** (links ←).
- Wiederhole** block: **solange** (wahr).
- mache** block: **zeichne Motiv** (male ein Motiv) with a 5x5 grid where the first column is checked. Coordinates: **bei x** (0), **y** (0).
- wiederhole** block: **4** mal.
- mache** block: **warte bis Taste ändert**.
- wenn** block: **Taste A** gedrückt.
- mache** block: **zeichne Motiv** (male ein Motiv) with a 5x5 grid where the first and second columns are checked. Coordinates: **bei x** (0), **y** (0).
- wenn** block: **Taste B** gedrückt.
- mache** block: **zeichne Motiv** (male ein Motiv) with a 5x5 grid where the first, second, and third columns are checked. Coordinates: **bei x** (0), **y** (0).
- warte (ms)** block: **250**.
- wiederhole** block: **8** mal.

# ARBEITSBLATT

## KOMPONENTENTEST

### Musterlösung (Teil 2 von 2)

The image shows a Scratch script for testing a component. It consists of several nested loops and conditional blocks. The script starts with a 'warte (ms)' block set to 250. It then enters a 'wiederhole' loop that repeats 8 times. Inside this loop, there is a 'zeichne Motiv' block that draws a 5x5 grid of pixels. The grid is initially empty, but as the script progresses, more pixels are drawn. The 'zeichne Motiv' block has a 'male ein Motiv' sub-block that sets the color of each pixel based on its coordinates (x and y). The 'wiederhole' loop is followed by a 'Wiederhole' loop that repeats 'solange' 'nicht' 'TasteGedrückt' is true. Inside this loop, there are four 'wenn' blocks that check if a specific 'Beinchen' (3, 2, 0, 1) is 'geerdet'. If it is, the script writes values to 'FarbanteilRot', 'FarbanteilGruen', 'FarbanteilBlau', and 'TasteGedrückt'. The values are 127, 0, 0, and 'wahr' for Beinchen 3; 0, 127, 0, and 'wahr' for Beinchen 2; 0, 0, 127, and 'wahr' for Beinchen 0; and 64, 64, 0, and 'wahr' for Beinchen 1. After the 'Wiederhole' loop, the script writes 'TasteGedrückt' to 'falsch', deletes the color pixels, and then counts 'LedIndex' from 0 to 11 with an increment of 1. For each 'LedIndex', it sets the color of the pixel at that index to 'rot', 'grün', or 'blau' based on the 'Farbanteil' values. Finally, it waits 250 ms and deletes the color pixels.

```
Scratch Script:
- warte (ms) 250
- wiederhole 8 mal
  - zeichne Motiv
    - male ein Motiv
      - 4 ✓
      - 3 ✓
      - 2
      - 1 ✓
      - 0 ✓
      - 0 1 2 3 4
    - bei x 0
    - y 0
  - Wiederhole solange nicht TasteGedrückt
    - mache
      - wenn Beinchen 3 geerdet
        - mache
          - Schreibe FarbanteilRot 127
          - Schreibe FarbanteilGruen 0
          - Schreibe FarbanteilBlau 0
          - Schreibe TasteGedrückt wahr
      - wenn Beinchen 2 geerdet
        - mache
          - Schreibe FarbanteilRot 0
          - Schreibe FarbanteilGruen 127
          - Schreibe FarbanteilBlau 0
          - Schreibe TasteGedrückt wahr
      - wenn Beinchen 0 geerdet
        - mache
          - Schreibe FarbanteilRot 0
          - Schreibe FarbanteilGruen 0
          - Schreibe FarbanteilBlau 127
          - Schreibe TasteGedrückt wahr
      - wenn Beinchen 1 geerdet
        - mache
          - Schreibe FarbanteilRot 64
          - Schreibe FarbanteilGruen 64
          - Schreibe FarbanteilBlau 0
          - Schreibe TasteGedrückt wahr
    - zeichne Motiv
      - male ein Motiv
        - 4
        - 3
        - 2 ✓
        - 1
        - 0
        - 0 1 2 3 4
      - bei x 0
      - y 0
    - Schreibe TasteGedrückt falsch
    - lösche Farbpixel
    - Zähle LedIndex von 0 bis 11 mit 1
    - mache
      - setze Farbpixel LedIndex
        - rot FarbanteilRot
        - grün FarbanteilGruen
        - blau FarbanteilBlau
      - warte (ms) 250
    - lösche Farbpixel
```

## ARBEITSBLATT

### KOMPONENTENTEST

#### RÜCKBLICK

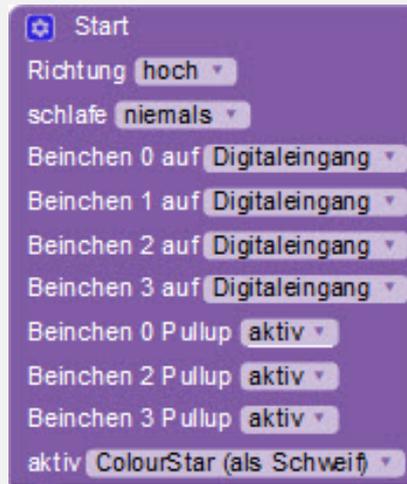
---

Wenn der Komponententest ohne Fehler durchläuft, dann bist du sicher, dass deine Leuchtscheibe-mini richtig funktioniert.

Viel Spass beim Realisierend deiner eigenen Ideen wünscht dir das Team von code+LIGHT.

### NEUE BEFEHLE

---



Im Startblock wird dem CodeBug mitgeteilt, wie er sich verhalten soll.

Für die vier externen farbigen Tasten müssen die vier Anschlüsse 0 bis 3 des CodeBug auf «Digitaleingang» eingestellt werden. So weiss der CodeBug, dass an diesen Anschlüssen entweder 0 V (tief) oder 5 V (hoch) anliegt und kann bei gedrückter Taste 0 V entsprechend als «Beinchen geerdet» erkennen.

Wenn die Tasten nicht gedrückt sind, soll 5 V am Beinchen 0 bis 3 anliegen dazu werden die internen Pullup-Widerstände aktiviert. Speziell ist, dass Beinchen 1 KEINEN internen Pullup-Widerstand hat. Daher wird er als zusätzliches Bauteil extern dazu gelötet. Dies ist im Zusammenbau der Leuchtscheibe-mini beschrieben.

#### WIE BIST DU VORGEANGEN?

---

1. Die LED-Matrix des CodeBug zeigt die beiden Muster an.
2. Jede der vier externen farbigen Tasten definiert ihre Farbe mit Nummern.
3. Damit der LED-Ring langsam mit Farbe gefüllt wird, hast du eine Verzögerung eingebaut.