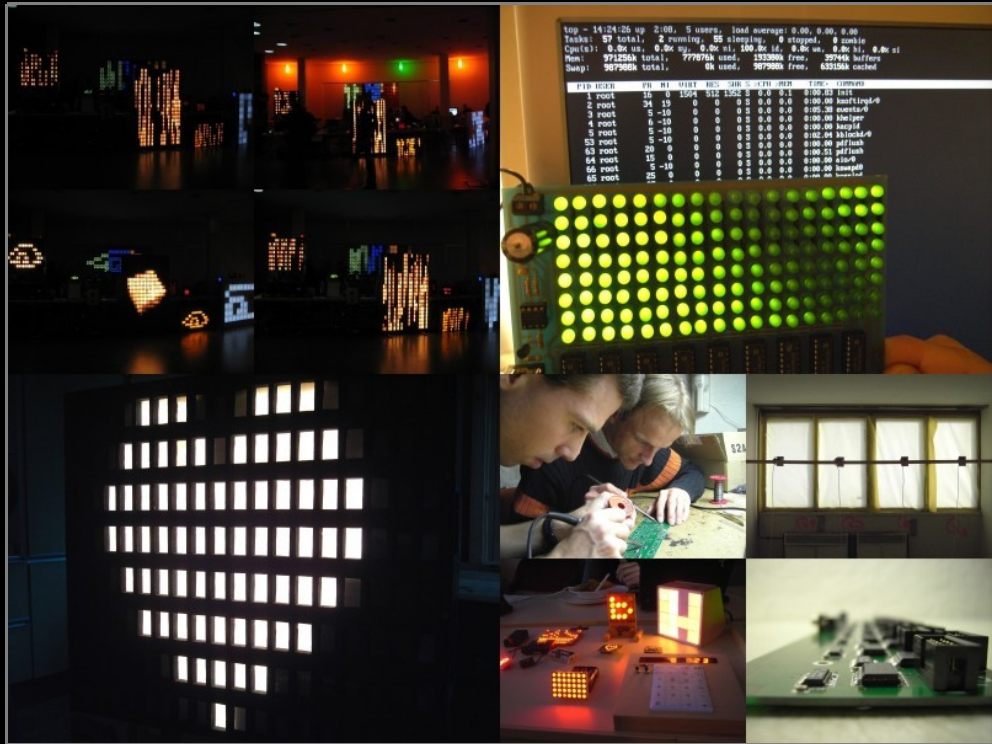


# Von der leuchtenden Idee zum blinkenden Haus



Stephan Kambor  
<st@blinkenarea.org>

Stefan Schürmans  
<stefan@blinkenarea.org>

Juliane Pilster  
<juliane@blinkenarea.org>

Chemnitzer Linux-Tage 2008

*Innovation dank Freiheit*



# Von der leuchtenden Idee zum blinkenden Haus

## 0. Inhalt

## 1. Entstehung der BlinkenArea

## 2. Nachbauprojekte

### 2.1. 144 Pixel

### 2.2. 520 Pixel

### 2.3. Vermischtes

## 3. Linux

### 3.1. Software

### 3.2. Formate, Protokolle

### 3.3. BlinkenLEDs

## 4. Eigenständige Projekte

### 4.1. TROIA

### 4.2. bluebox

## 5. Stand der Dinge / Ausblick

### 5.1. Allgemein

### 5.2. Spendenaktionen

### 5.3. Ausstellungen

### 5.4. Ausblick

## 6. Ende

# Idee der BlinkenArea: Blinkenlights

- 2001: Blinkenlights



- 2004: Blinkenlights Reloaded

- 2002: Blinkenlights Arcade



# Idee der BlinkenArea: Entstehung

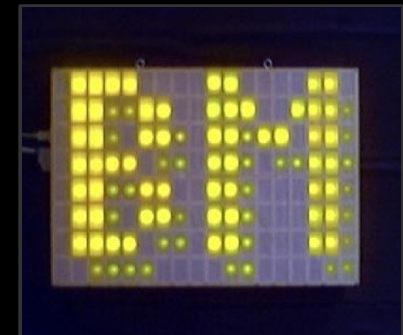
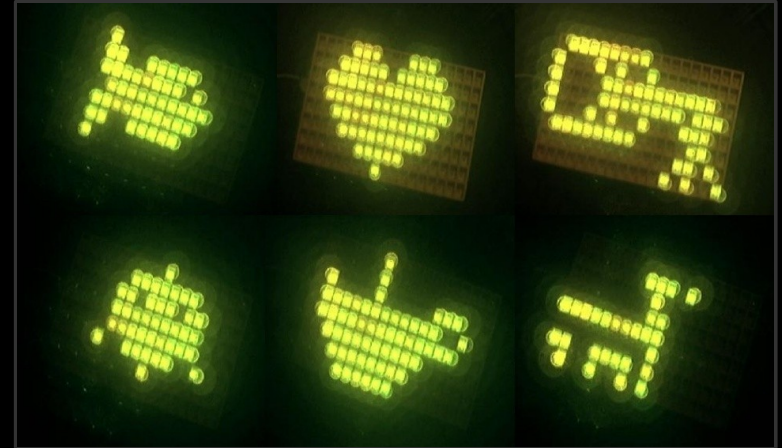
- 2003: 2. Chaos Communication Camp
- erstes Treffen der Teilnehmer, die an verschiedenen blinkenden Projekten arbeiteten
- Erfolg der BlinkenArea bei den Camp-Besuchern
- Plattform für Unterstützung, Zusammenarbeit und Austausch





# Nachbauprojekte: BLINKENmini

- erstes Blinkenlights Nachbauprojekt
- 73 cm breit x 49 cm hoch
- 18x8 Pixel
- superhelle 10mm LEDs, gelb, 50 mA
- 8 Graustufen
- 4 x SNES Pad
- Serielle Schnittstelle
- Life, Pong (4), Snake (1), Tetris (1), Tron (4)



<http://wiki.blinkenarea.org/wiki/BLINKENmini>

# Nachbauprojekte: LittleLights

- Blinkenlights-Nachbau im Maßstab 1:42
- 75 cm breit, 115 cm hoch
- 18x8 Pixel
- Glühlampen E10, 36V, 1,8 W
- s/w - 8 Graustufen
- serielle Schnittstelle



<http://wiki.blinkenarea.org/wiki/LittleLights>

# Nachbauprojekte: BlinkenLEDs

- Konzept / Platine
- flexibel einsetzbar
- 18x8 Pixel
- Low current LEDs
- monochrom
- Parallele Schnittstelle
- Graustufen mit Shifter oder BlinkenLEDsPro



<http://wiki.blinkenarea.org/wiki/BlinkenLEDs>



# Nachbauprojekte: XMasLights

- 288 Lichterkettenlämpchen
- 18x8 Pixel
- monochrom
- ATtiny Prozessor / Schieberegister / Transistoren
- serielle Schnittstelle
- eingebauter Flash-EEPROM

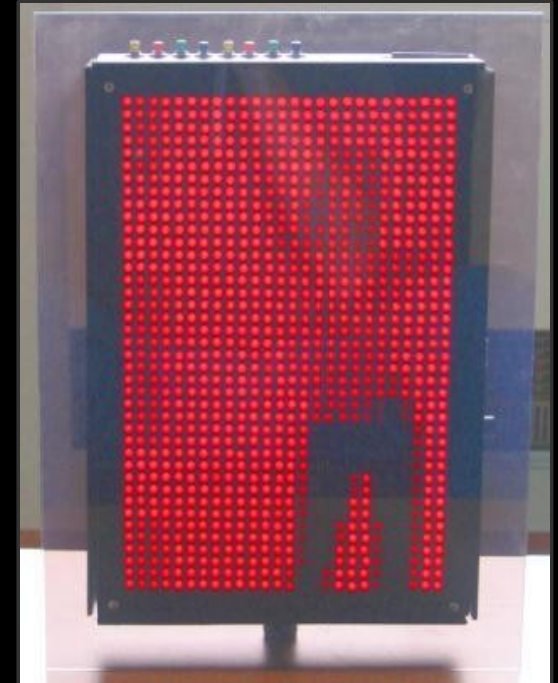
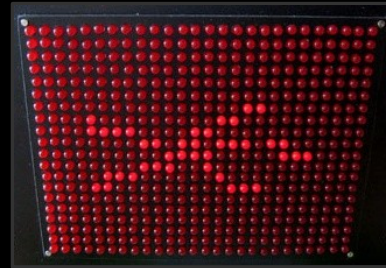
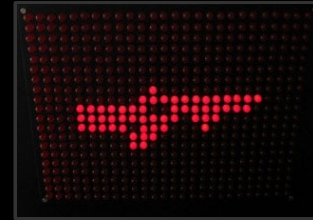


<http://wiki.blinkenarea.org/wiki/XMasLights>



# Nachbauprojekte: ARCADEmini

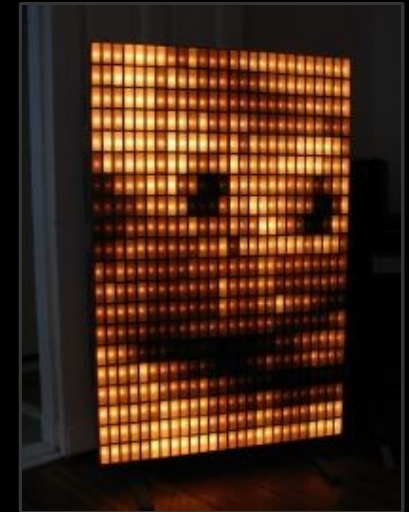
- 520 / 1040 Low current LEDs
- 20x26 Pixel
- 16 Graustufen
- Standalone
- USB / Compact Flash
- PIC18F8520 Prozessor
- 8 Knöpfe + 2 SNES-Pads
- programmierbar (ARCADEmini-Assembler)
- Pacman, Tetris, Pong, Dezimal- / BCD- / Binär-Uhr, ...



<http://wiki.blinkenarea.org/wiki/ARCADEmini>

# Nachbauprojekte: ArcadeMaxi (Glühbirnenglotze)

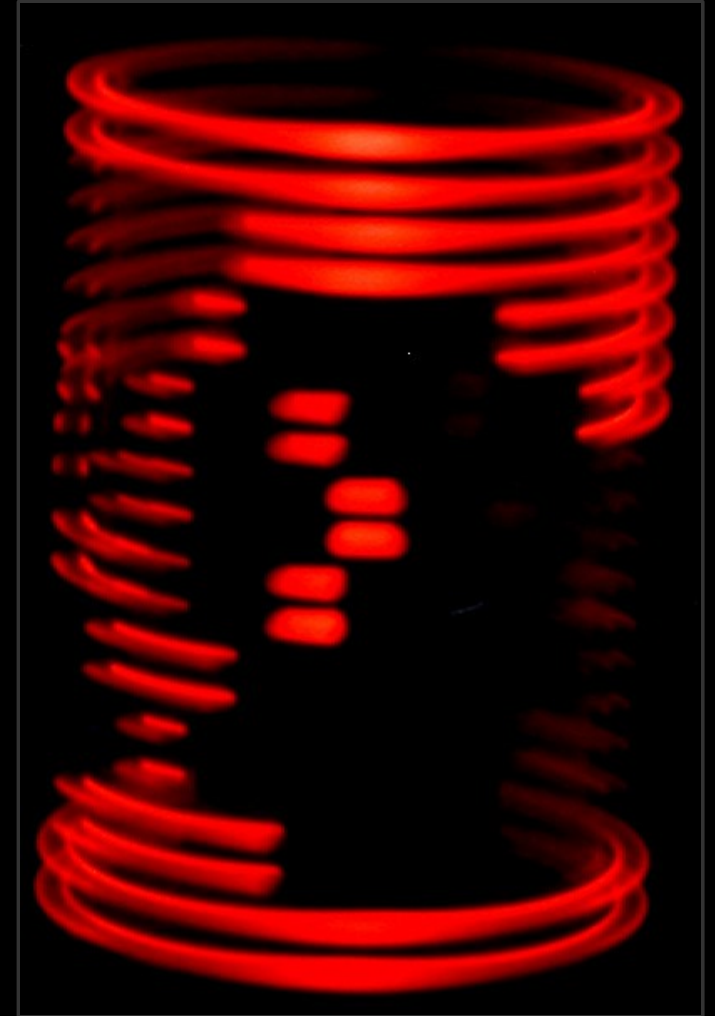
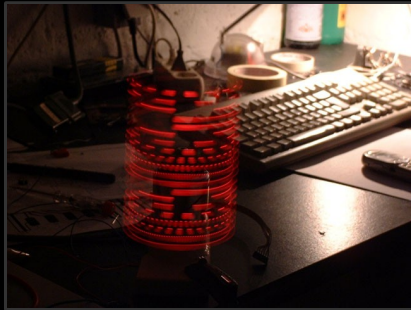
- Basis ARCADEmini
- 520 E 10 Glühlämpchen, 18 V, 100 mA
- 98,4 x 149,6 cm
- 16 Graustufen
- Selbstbau-Netzteil 20 V 40 A



<http://wiki.blinkenarea.org/wiki/ArcadeMaxi>

# Nachbauprojekte: RotArcade

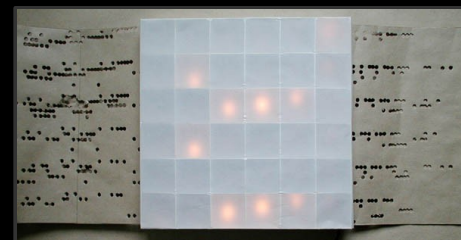
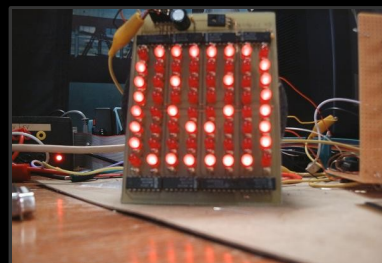
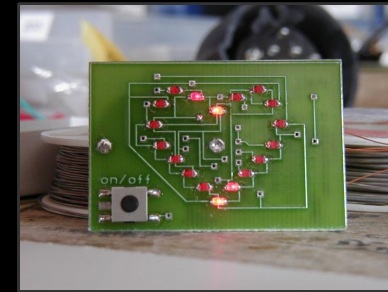
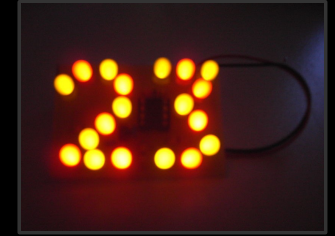
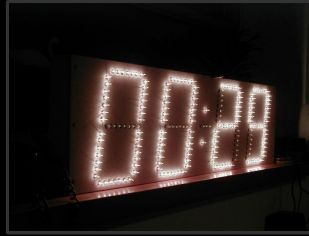
- Propeller-Clock-Prinzip:
  - 20 LEDs rotieren mit 3000 U/min
- 8 Graustufen
- Filme im 128K Flash-EEPROM gespeichert
- IR-Verbindung zum PC



<http://wiki.blinkenarea.org/wiki/RotArcade>

# Projekte: Vermischtes

- LED-23
- AntiKippenLights
- BlinkenBadge
- BlinkenWall
- C64Lights
- Ornament
- OrnaMatrix
- PapierPixel
- MovieMatrix
- MegaClock

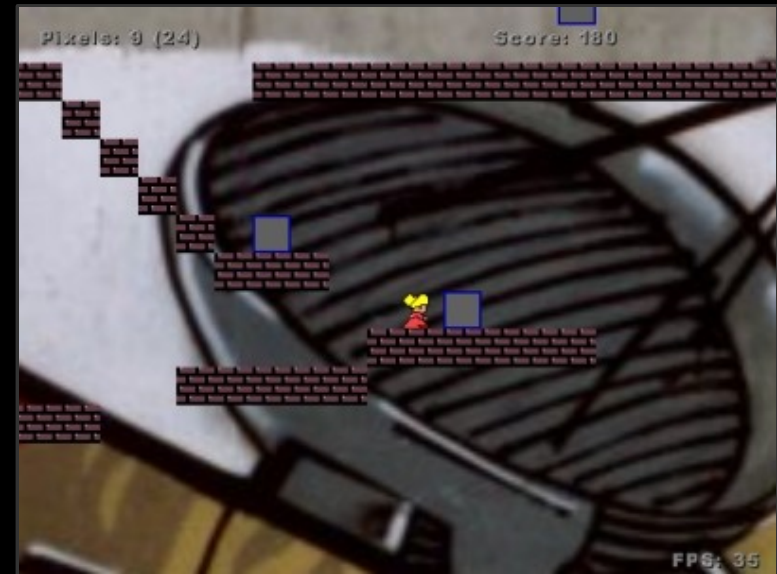


<http://wiki.blinkenarea.org/wiki/Projekte>



# Projekte: BlinkenSisters - Hunt for the Lost Pixels

- OpenSource-Jump'n'Run
- Plattformübergreifend
- eigene Level erstellen
- Joystick / Pad support



*Agent Kate, Agentin der geheimen BlinkenArea-Special-Forces muss in der Dimension B15 versprengte Pixel wieder finden und es mit Datenkraken, Augenmonstern und anderem "Getier" aufnehmen ...*

<http://wiki.blinkenarea.org/wiki/BlinkenSisters>

# Linux / Open Source + Blinken\*

Blinken\*-Software basiert auf Linux und Open Source

- alles dokumentiert, keine Geheimnisse
- keine Kosten für Software
  - Betriebssystem
  - Compiler
  - fertige Blinken\*-Programme
- Zusammenarbeit der Blinken\*-Gemeinde
  - Wiederverwendung von Software anderer Projekte
  - offene Dateiformate / Netzwerkprotokolle



# Blinkenlights Chaos Control Center - blccc

zentrales Steuerprogramm für Blinken\*

- Features
  - Abspielen von Filmen nach Playlist
  - Spiele und Effekte aus Modulen
  - Starten von Filmen und Spielen per Telefon
  - Ausgabe der Bilder als Stream über UDP/IP
- Open Source von „Project Blinkenlights“
  - geschrieben für Blinkenlights, Blinkenlights Arcade
  - Weiterentwicklung für bluebox durch BlinkenArea
  - Mirror: <http://stefan.blinkenarea.org/blinkenlights/>

# Blinkenlights ISDN - blinkisdn

Schnittstelle zwischen ISDN und blccc

- Features
  - Annahme von Anrufen auf ISDN-Leitungen
  - Abspielen eines Ansagetextes
  - Erkennung der DTMF-Töne (gedrückte Tasten)
- Open Source von „Project Blinkenlights“
  - geschrieben für Blinkenlights, Blinkenlights Arcade
  - Verwendung bei bluebox
  - Mirror: <http://stefan.blinkenarea.org/blinkenlights/>



# BlinkenLib

## Tools für Blinken\*

- kleine C-Library ohne Abhängigkeiten
  - Dateiformate
  - Netzwerkprotokolle
- einfache Tools
  - Konverter für Dateiformate
  - Senden/Empfangen von Streams
  - Ausgabe von Streams an Hardware
- Open Source von der BlinkenArea

<http://stefan.blinkenarea.org/BlinkenLib>

# Blinkenlights Proxy - bl\_proxy

## zentraler Stream-Manager

- Empfang von Streams
  - Filterung nach Quelle, Protokoll und Format
- Auswahl von Streams
  - Erkennung aktiver und inaktiver Streams
  - Auswahl gemäß Priorität
- Ausgabe von Streams
  - statisch und dynamisch
- Open Source von der BlinkenArea

[http://stefan.blinkenarea.org/bl\\_proxy](http://stefan.blinkenarea.org/bl_proxy)

# Dateiformate

- BlinkenLights Movie (BLM)  
s/w, Text-basiert
- BlinkenMini Movie (BMM)  
Graustufen, Text-basiert
- Blinkenlights Markup Language (BML)  
Graustufen / Farbe, XML-basiert
- Binary Blinken Movie (BBM)  
Graustufen / Farbe, binär

<http://wiki.blinkenarea.org/wiki/Dateiformate>

# BlinkenLights Movie (BLM)

einfache Textdatei

- für schwarz/weiß-Filme
- Erkennungszeile am Anfang
- jedes Bild einzeln (Anzeigedauer in Millisekunden)
- Pixeldaten unkomprimiert
  - zeilenweise
  - von links oben nach rechts unten
  - 0=aus, 1=an



# BlinkenLights Movie - Beispiel

# BlinkenLights Movie 18x8

# Beispielfilm

# zeigt ein Herz

@200

```
00000000000000000000
000011100011100000
000111110111110000
000111111111110000
000011111111100000
000000111110000000
000000001000000000
000000000000000000
```

@800

```
000011100011100000
```

...

...

```
111111111111111111
```

@800

```
111100011100011111
111000001000001111
110000000000000111
110000000000000111
111000000000000111
111100000000011111
111111000001111111
111111110111111111
```

# BlinkenMini Movie (BLM)

einfache Textdatei

- Erkennungszeile am Anfang
- für Graustufen-Filme
- jedes Bild einzeln (Anzeigedauer in Millisekunden)
- Pixeldaten unkomprimiert
  - zeilenweise
  - von links oben nach rechts unten
  - 0x00=aus, ..., 0xFF=voll an

# BlinkenMini Movie - Beispiel

# BlinkenMini Movie 18x8

# Beispielfilm

# zeigt ein Herz

@200

```
0x00 0x00 0x00 0x00 0xFF 0xFF 0xFF 0x00 0x00 0x00 0xFF 0xFF ...
0x00 0x00 0x00 0xFF 0x66 0x66 0x66 0xFF 0x00 0xFF 0x66 0x66 ...
0x00 0x00 0xFF 0x66 0x66 0x66 0x66 0x66 0xFF 0x66 0x66 0x66 ...
0x00 0x00 0xFF 0x66 0x66 0x66 0x66 0x66 0x66 0x66 0x66 0x66 ...
0x00 0x00 0x00 0xFF 0x66 0x66 0x66 0x66 0x66 0x66 0x66 0x66 ...
0x00 0x00 0x00 0x00 0xFF 0xFF 0x66 0x66 0x66 0x66 0x66 0xFF ...
0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0xFF 0x66 0xFF 0xFF 0x00 ...
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x00 0x00 0x00 ...
```

@800

```
0x11 0x11 0x11 0x11 0xEE 0xEE 0xEE 0x11 0x11 0x11 0xEE 0xEE ...
```

...

# Blinkenlights Markup Language (BML)

## XML-basiert

- für Graustufen- und Farb-Filme
- Header mit allgemeinen Infos
- jedes Bild einzeln (Anzeigedauer in Millisekunden)
- Pixeldaten unkomprimiert
  - zeilenweise
  - von links oben nach rechts unten
  - Pixel = Werte aller Kanäle
  - z.B. für 8 Bit pro Kanal: 00=aus, ..., FF=an



# Blinkenlights Markup Language - Beispiel

```
<blm height="8" width="18" channels="1" bits="8">
  <header>
    <title>Film mit Herz</title>
    <author>Herz-Blinker</author>
  </header>

  <frame duration="200">
    <row>00000000FFFFFF000000FFFFFF00000000</row>
    <row>000000FF696969FF00FF696969FF000000</row>
    <row>0000FF6969696969FF6969696969FF0000</row>
    <row>0000FF6969696969696969696969FF0000</row>
    <row>000000FF696969696969696969FF000000</row>
    <row>00000000FFFF6969696969FFFF00000000</row>
    <row>000000000000FFFF69FFFF000000000000</row>
    <row>0000000000000000FF0000000000000000</row>
  </frame>

  ...

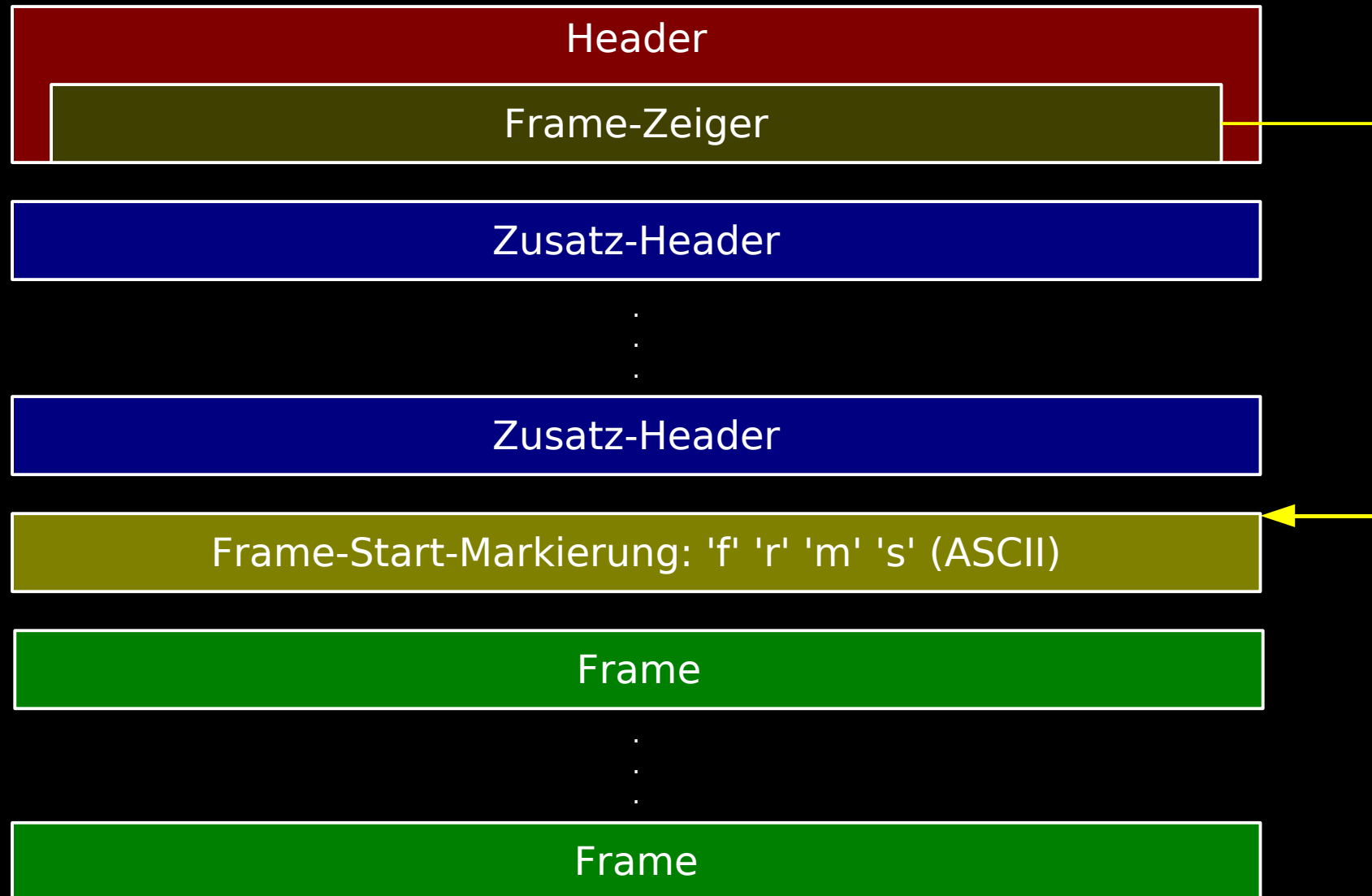
</blm>
```

# Binary Blinken Movie (BBM)

## Binärformat

- für Graustufen- und Farb-Filme
- Header mit Format
- allgemeine Infos
- jedes Bild einzeln (Anzeigedauer in Millisekunden)
  - Pixeldaten unkomprimiert, binär
  - zeilenweise
  - von links oben nach rechts unten
  - Pixel = Werte aller Kanäle

# Binary Blinken Movie - Struktur



# Binary Blinken Movie - Header



# Binary Blinken Movie - Zusatz-Header

## Zusatz-Header

Kennung (je nach Zusatz-Header)

Länge des Zusatz-  
Headers

## Zusatz-Header: Info

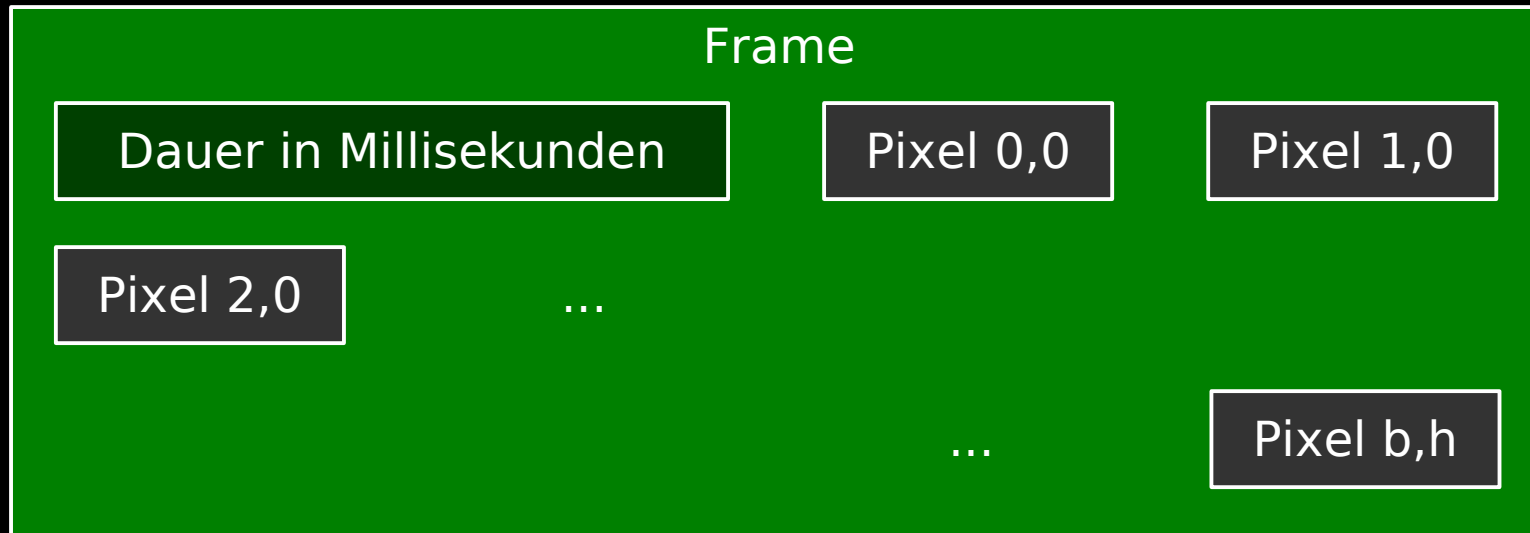
Kennung: 'i' 'n' 'f' 'o' (ASCII)

Länge des Zusatz-  
Headers

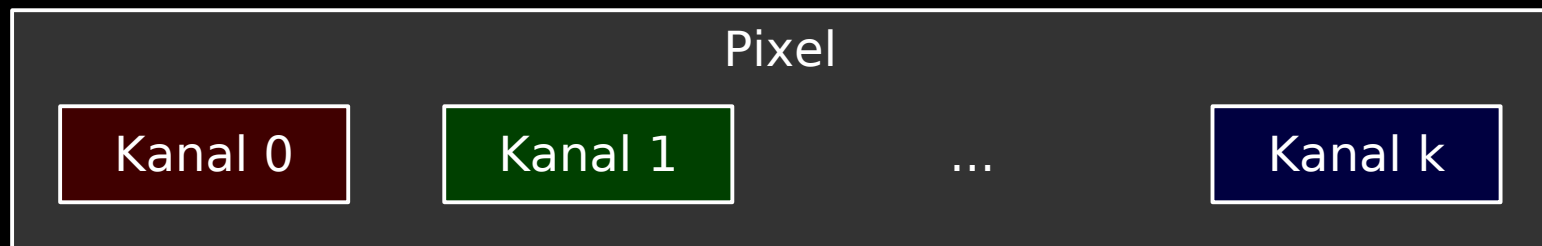
Info-Typ: 0-term. String (z.B. „author“)

Info-Inhalt: 0-term. String (z.B. „Herz-Blinker“)

# Binary Blinken Movie - Frame



$b := \text{Breite} - 1, h := \text{Höhe} - 1$



$k := \text{Anzahl Kanäle} - 1, \text{Kanal} := 0x00..\text{Maximalwert}$



# Netzwerkprotokolle

- BlinkenLights Protocol (BLP)  
s/w, UDP/IP
- Extended BlinkenLights Protocol (EBLP)  
Graustufen, UDP/IP
- Microcontroller Unit Frame (MCUF)  
Graustufen / Farbe, UDP/IP

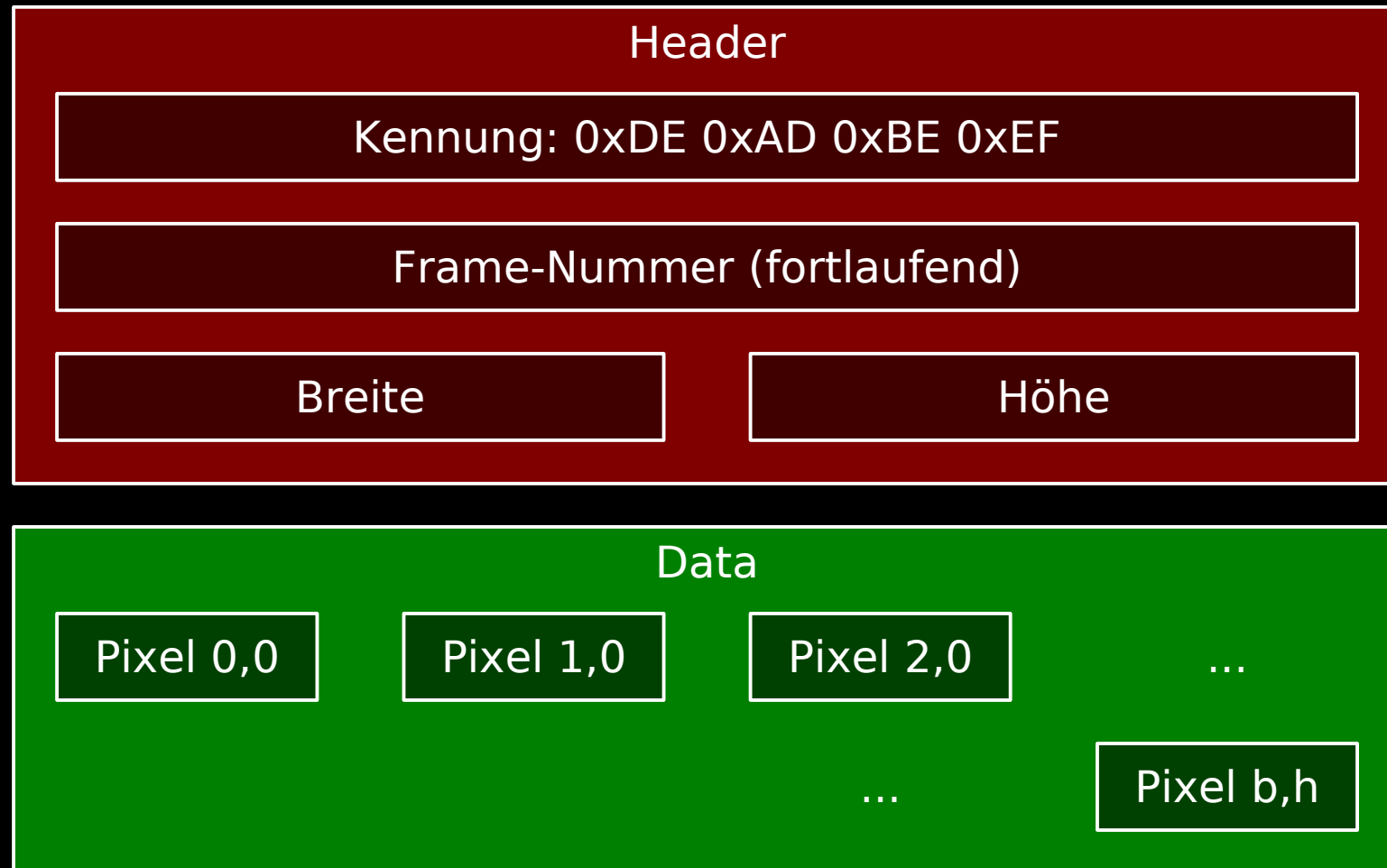
<http://wiki.blinkenarea.org/wiki/Protokolle>

# BlinkenLights Protocol (BLP)

UDP/IP, 1 Frame = 1 Paket

- für schwarz/weiß-Filme
- Header mit Format
- Pixeldaten unkomprimiert, binär
  - zeilenweise
  - von links oben nach rechts unten
  - 0x00=aus, 0x01=an
- Dauer der Frames nur implizit durch zeitlichen Abstand der Pakete

# BlinkenLights Protocol - Frame



$b := \text{Breite} - 1$ ,  $h := \text{Höhe} - 1$ , Pixel := 0x00/0x01

# BlinkenLights Protocol - dynamisch

Stream-Anforderung durch Empfänger

- optional
- Kommando Paket zur Stream-Anforderung
  - vom Empfänger gesendet
  - alle 10 Sekunden wiederholt
- Kommando-Paket zur Aufhebung der Stream-Anforderung
  - vom Empfänger gesendet
  - alternativ: Ende durch Timeout

# BlinkenLights Protocol – Kommandos

## Stream-Anforderung

Kennung: 0xDE 0xAD 0xBE 0xCD

Kommando: 'R' 'E' 'F' 'R' 'E' 'S' 'H' (ASCII)

## Aufhebung der Stream-Anforderung

Kennung: 0xDE 0xAD 0xBE 0xCD

Kommando: 'C' 'L' 'O' 'S' 'E' (ASCII)

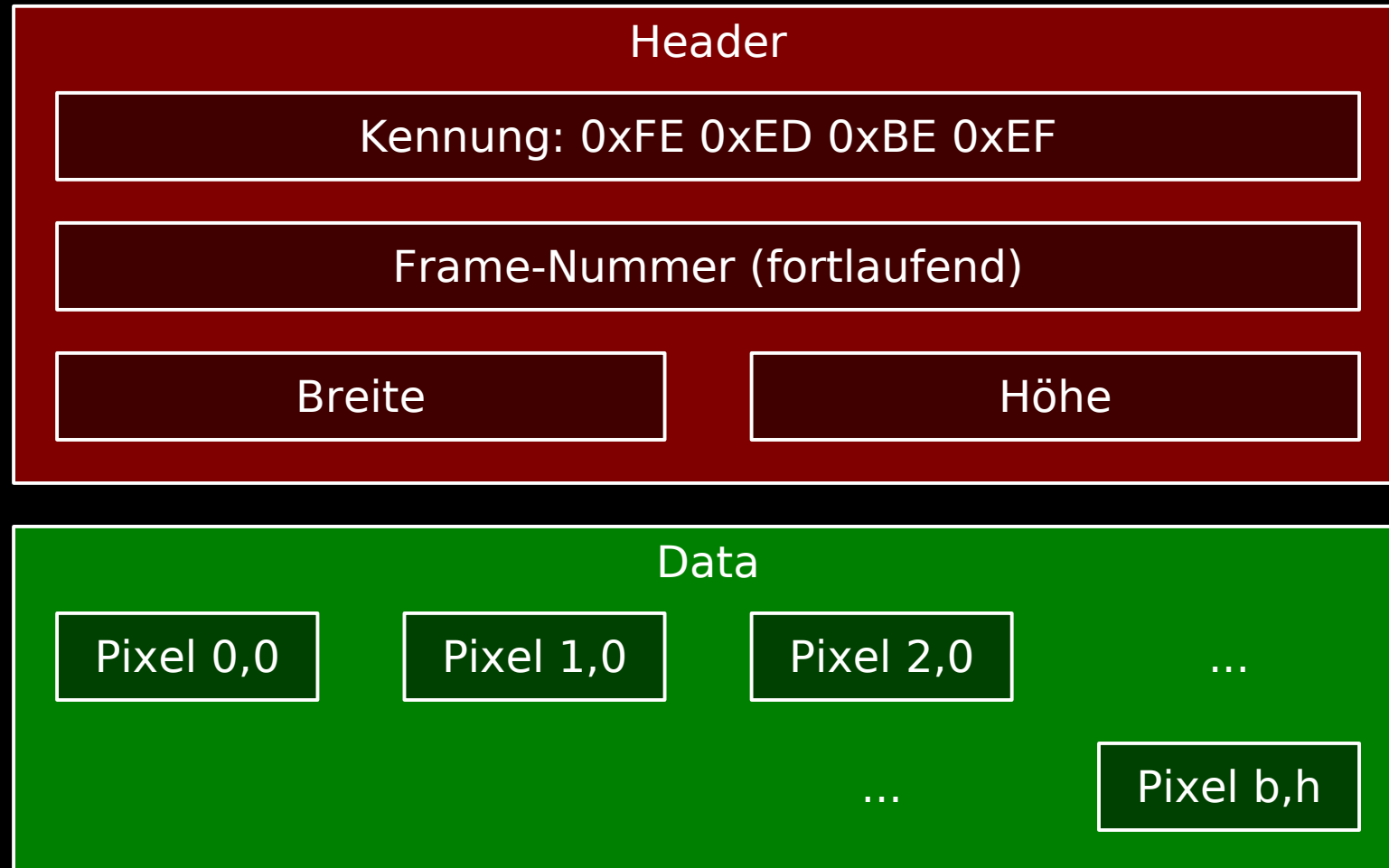
# Extended BlinkenLights Protocol (EBLP)

UDP/IP, 1 Frame = 1 Paket

- für Graustufen-Filme
- Header mit Format
- Pixeldaten unkomprimiert, binär
  - zeilenweise
  - von links oben nach rechts unten
  - 0x00=aus, ..., 0xFF=voll an
- Dauer der Frames nur implizit durch zeitlichen Abstand der Pakete



# Extended BlinkenLights Protocol - Frame



$b := \text{Breite} - 1$ ,  $h := \text{Höhe} - 1$ , Pixel := 0x00..0xFF

# Extended BlinkenLights Protocol - dynamisch

Stream-Anforderung durch Empfänger

- optional
- Kommando Paket zur Stream-Anforderung
  - vom Empfänger gesendet
  - alle 10 Sekunden wiederholt
- Kommando-Paket zur Aufhebung der Stream-Anforderung
  - vom Empfänger gesendet
  - alternativ: Ende durch Timeout

# Extended BlinkenLights Protocol - Kommandos

## Stream-Anforderung

Kennung: 0xDE 0xAD 0xBE 0xCD

Kommando: 'R' 'E' 'F' 'R' 'E' 'S' 'H' '2' '5' '6' (ASCII)

## Aufhebung der Stream-Anforderung

Kennung: 0xDE 0xAD 0xBE 0xCD

Kommando: 'C' 'L' 'O' 'S' 'E' '2' '5' '6' (ASCII)

# Microcontroller Unit Frame (MCUF)

UDP/IP, 1 Frame = 1 Paket

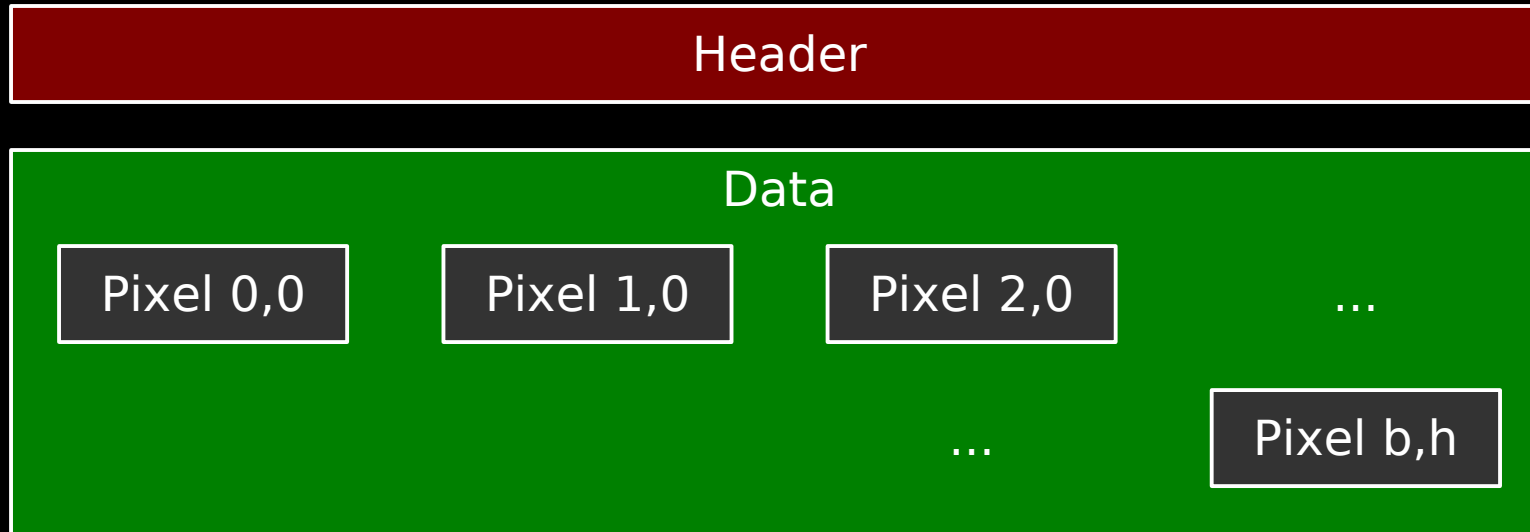
- für Graustufen- und Farb-Filme
- Header mit Format
- Pixeldaten unkomprimiert, binär
  - zeilenweise
  - von links oben nach rechts unten
  - Pixel = Werte aller Kanäle
- Dauer der Frames nur implizit durch zeitlichen Abstand der Pakete

# Microcontroller Unit Frame – Frame (1/2)

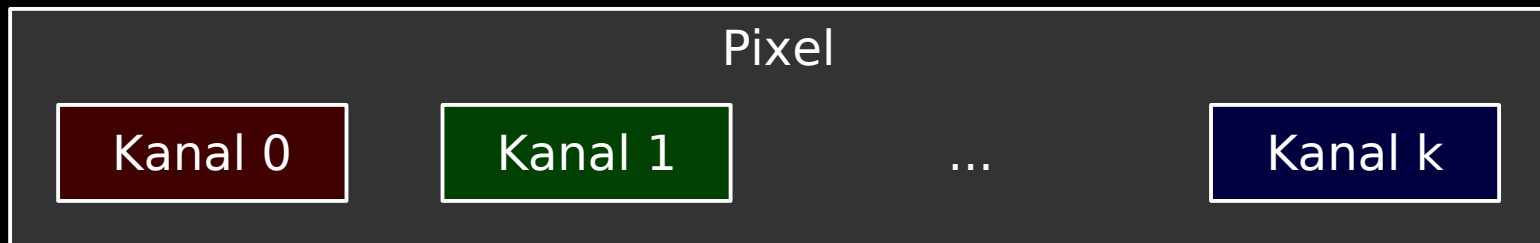


$b := \text{Breite} - 1, h := \text{Höhe} - 1$

# Microcontroller Unit Frame – Frame (2/2)



$b := \text{Breite} - 1, h := \text{Höhe} - 1$



$k := \text{Anzahl Kanäle} - 1, \text{Kanal} := 0x00..\text{Maximalwert}$



# Microcontroller Unit Frame – dynamisch

Stream-Anforderung durch Empfänger

- optional
- Kommando Paket zur Stream-Anforderung
  - vom Empfänger gesendet
  - alle 10 Sekunden wiederholt
- Kommando-Paket zur Aufhebung der Stream-Anforderung
  - vom Empfänger gesendet
  - alternativ: Ende durch Timeout

# Microcontroller Unit Frame – Kommandos

## Stream-Anforderung

Kennung: 0x42 0x42 0x42 0x42

reserviert: 0x00 0x00 0x00 0x00 0x00 0x00

## Aufhebung der Stream-Anforderung

Kennung: 0x42 0x42 0x42 0x43

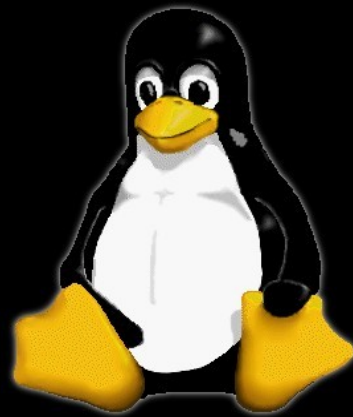
reserviert: 0x00 0x00 0x00 0x00 0x00 0x00

# Graustufen mit Linux



schwarz/weiss

+



=



Graustufen

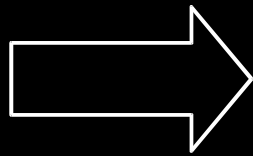
# BlinkenLEDs-Prinzip

18x8 Pixel, schwarz/weiss

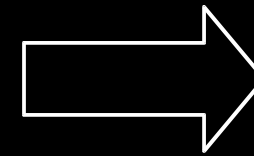


1 Parallelport

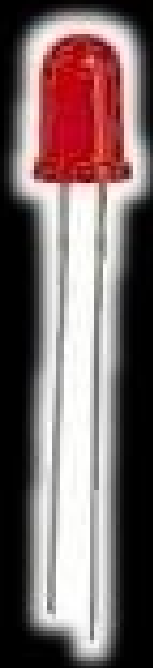
9 Datenleitungen  
2 Steuerleitungen



18 Schieberegister  
mit Latch

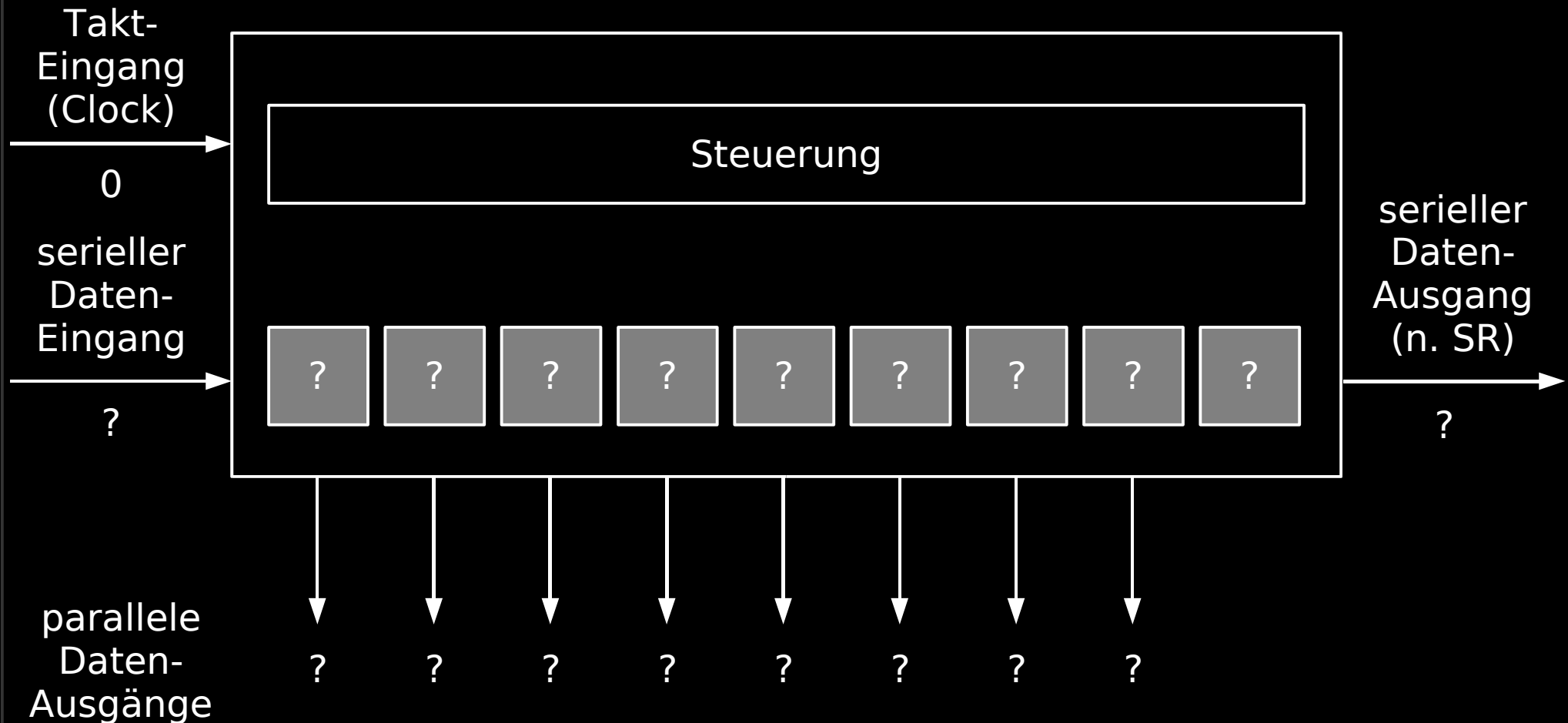


144 Ausgänge

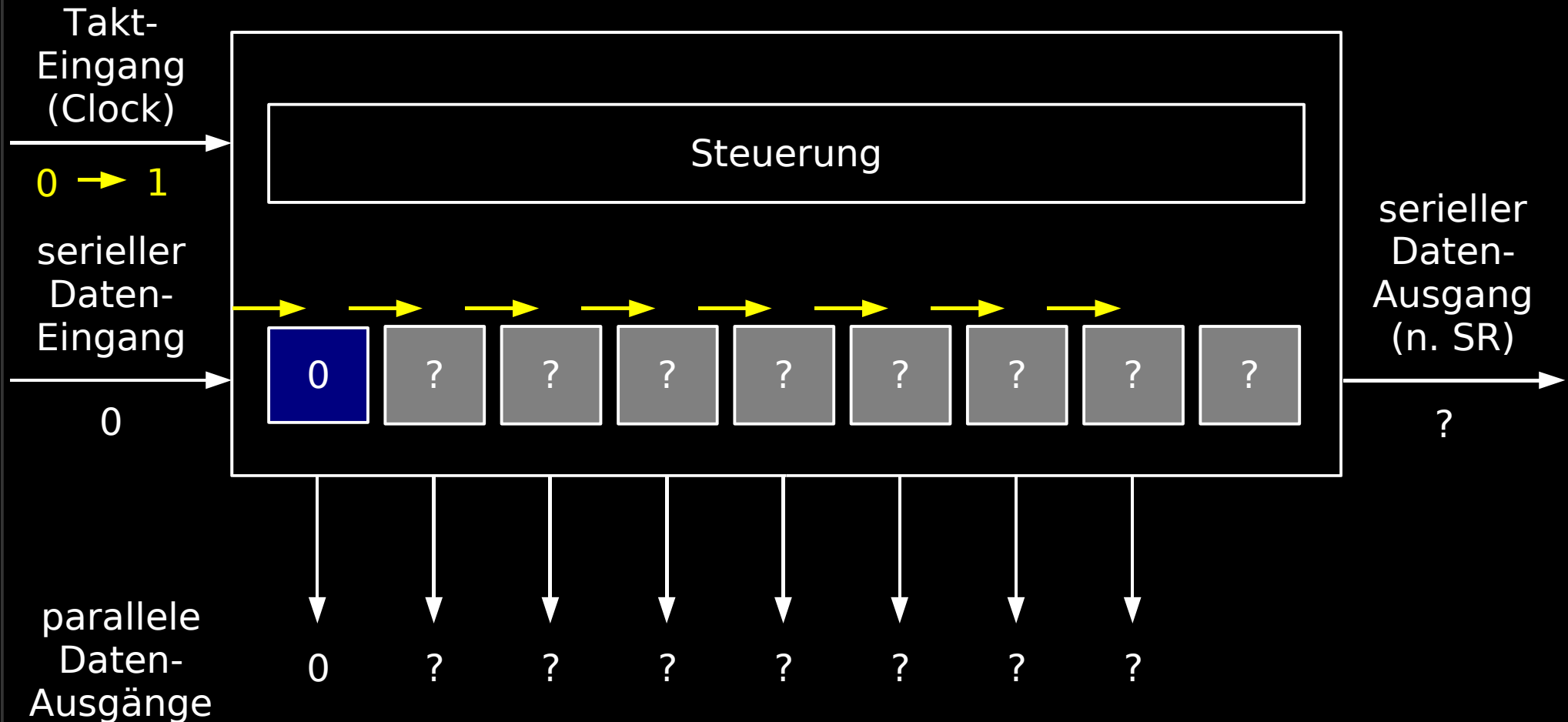


144 LEDs

# Schieberegister (SR)

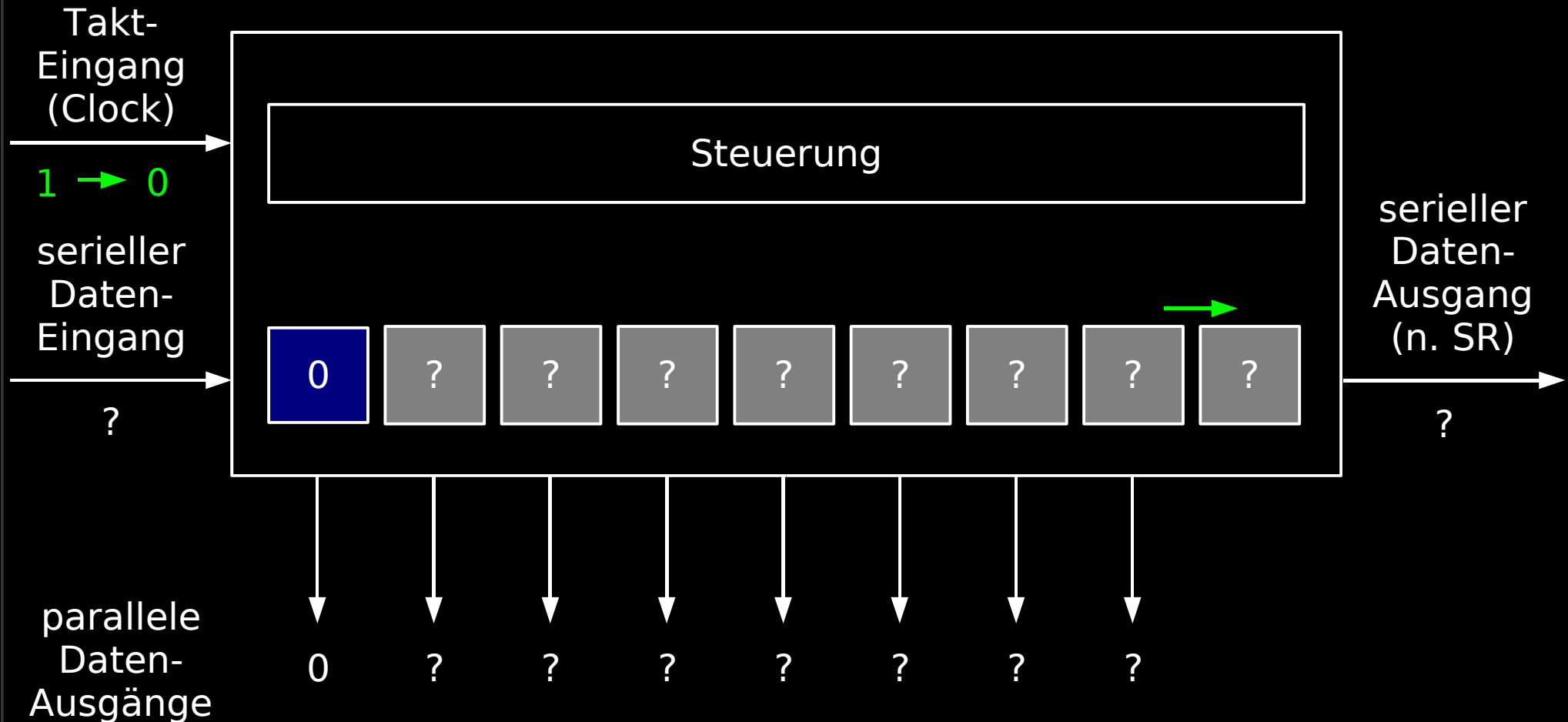


# Schieberegister – pos. Flanke

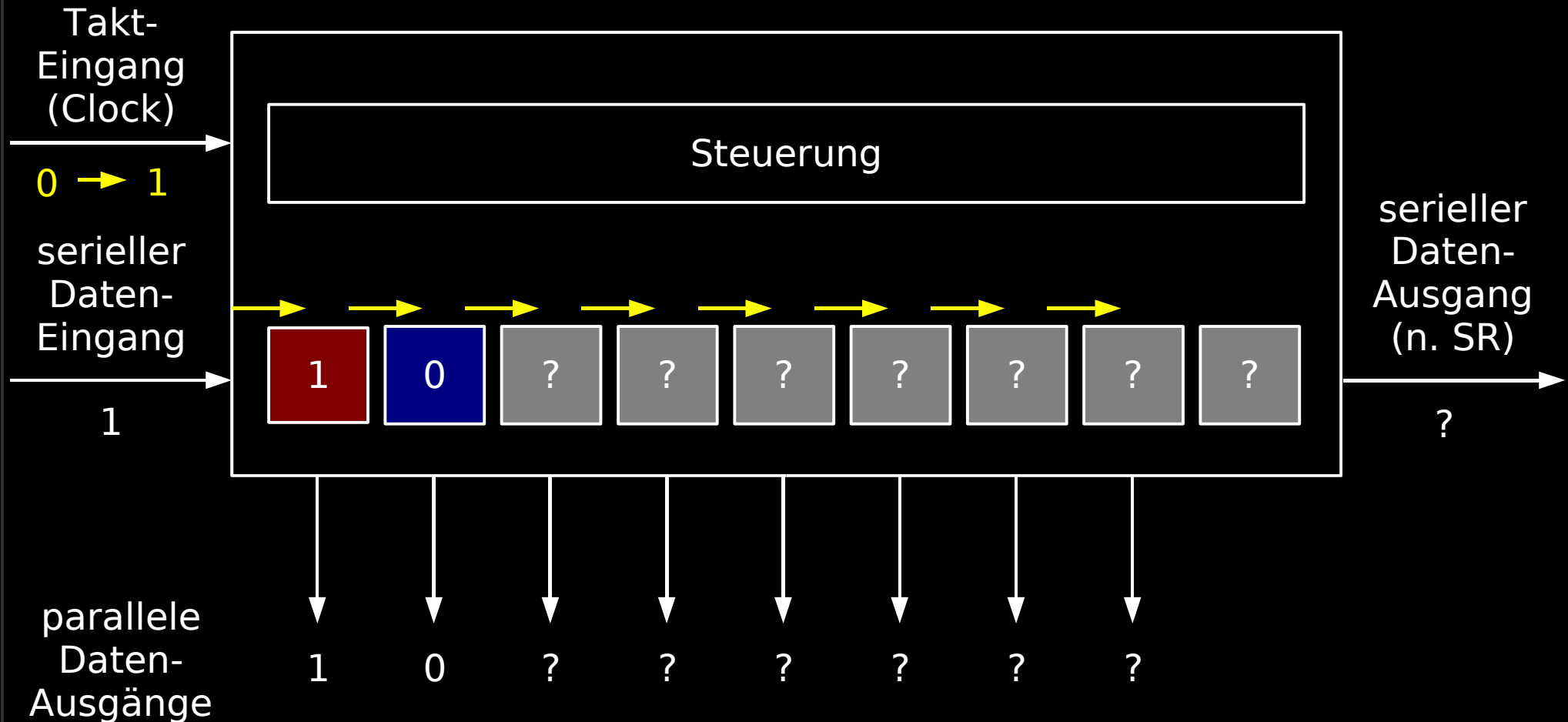




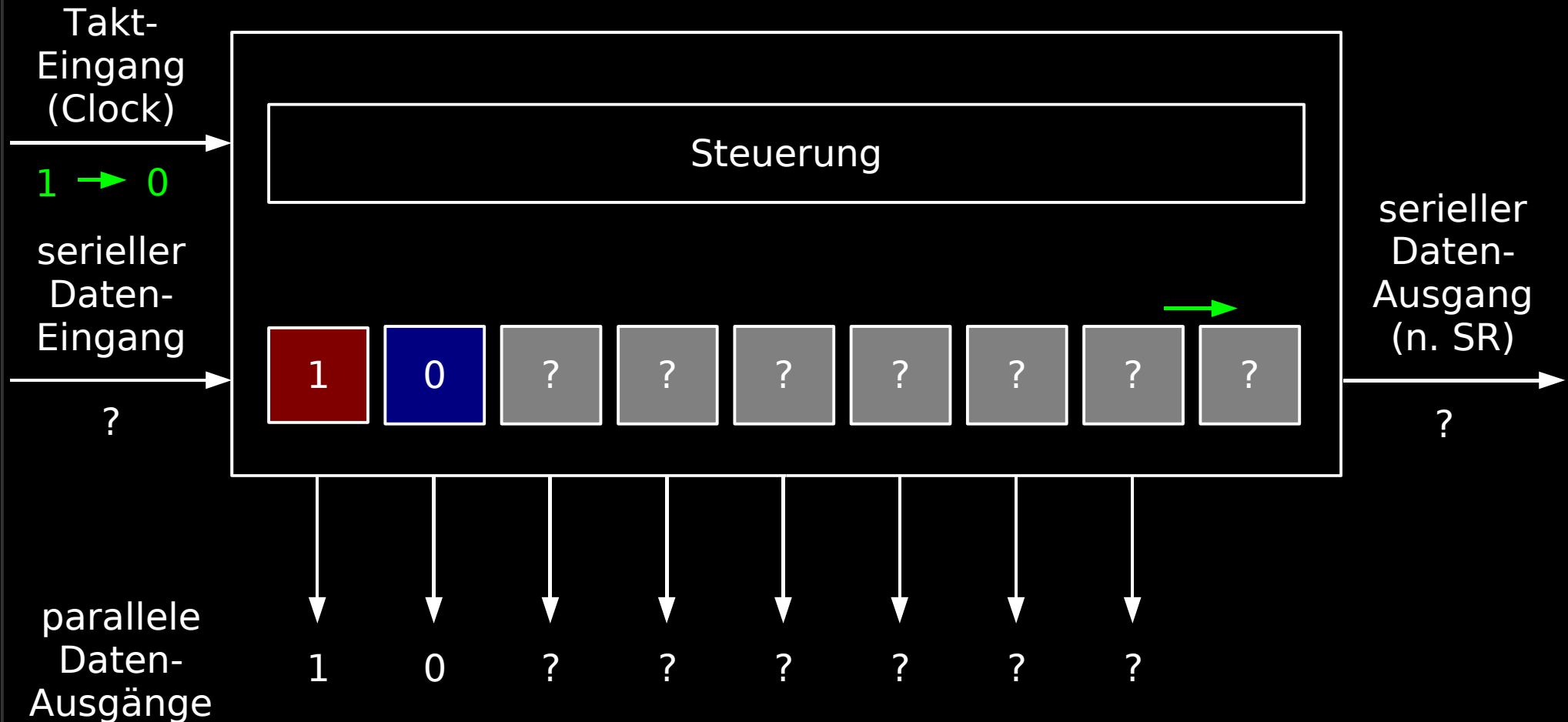
# Schieberegister – neg. Flanke



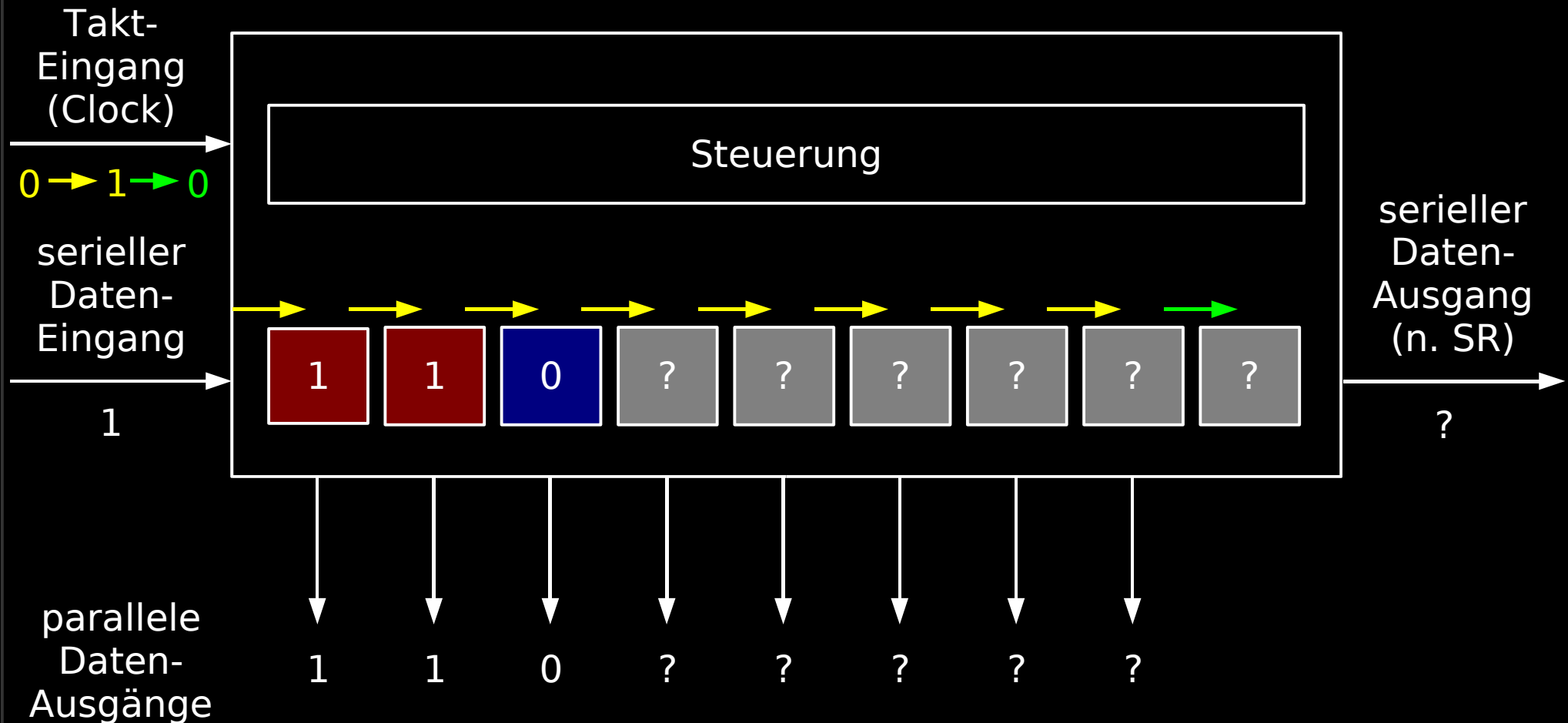
# Schieberegister – 2. pos. Flanke



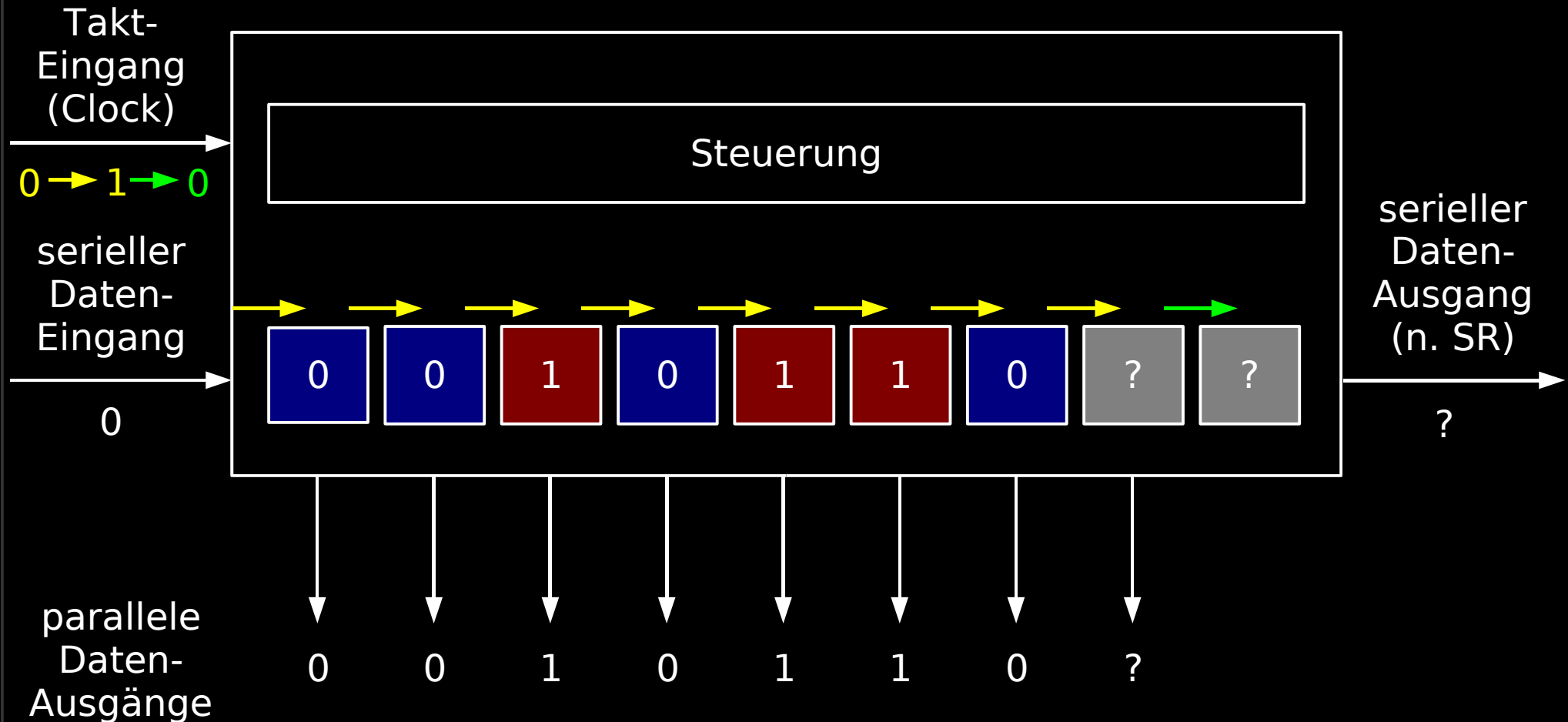
# Schieberegister – 2. neg. Flanke



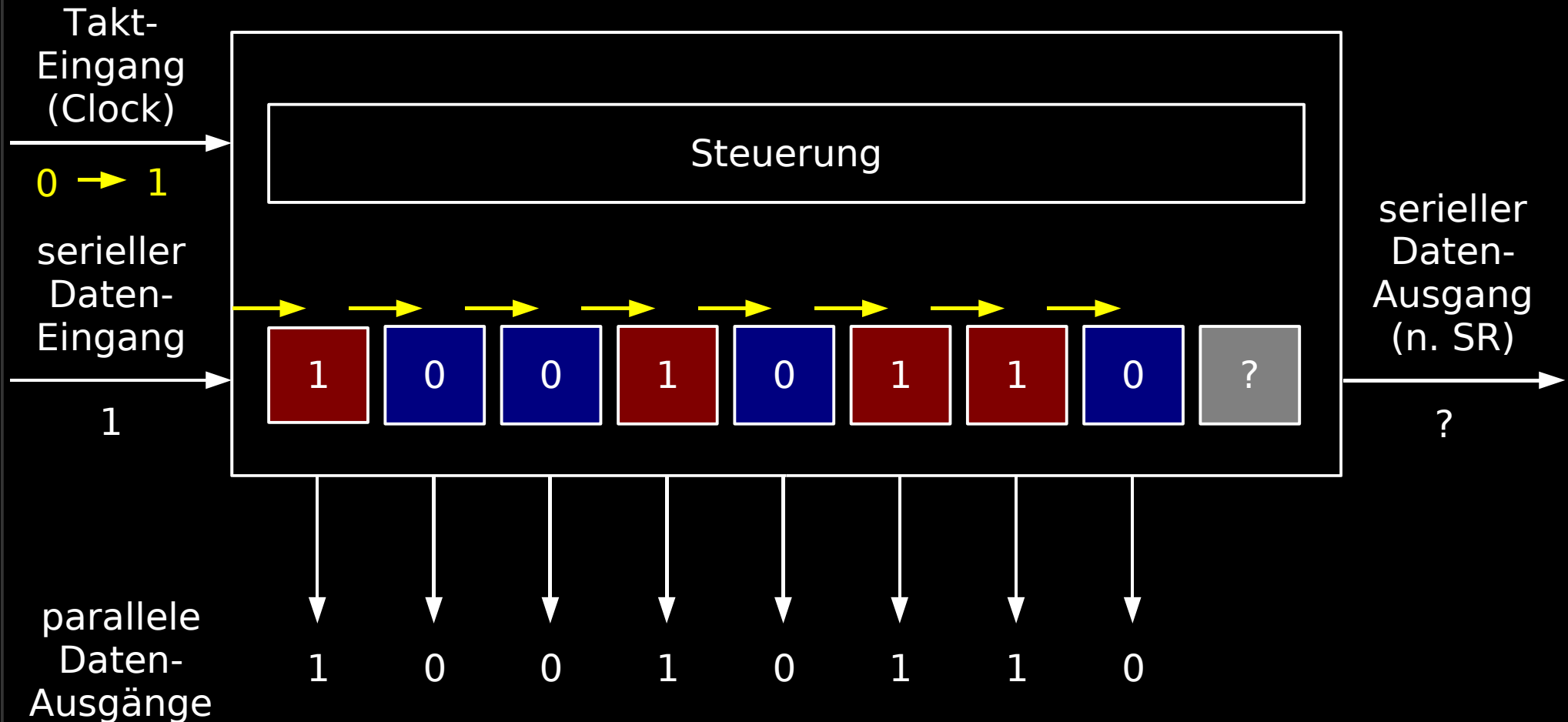
# Schieberegister – 3. Bit



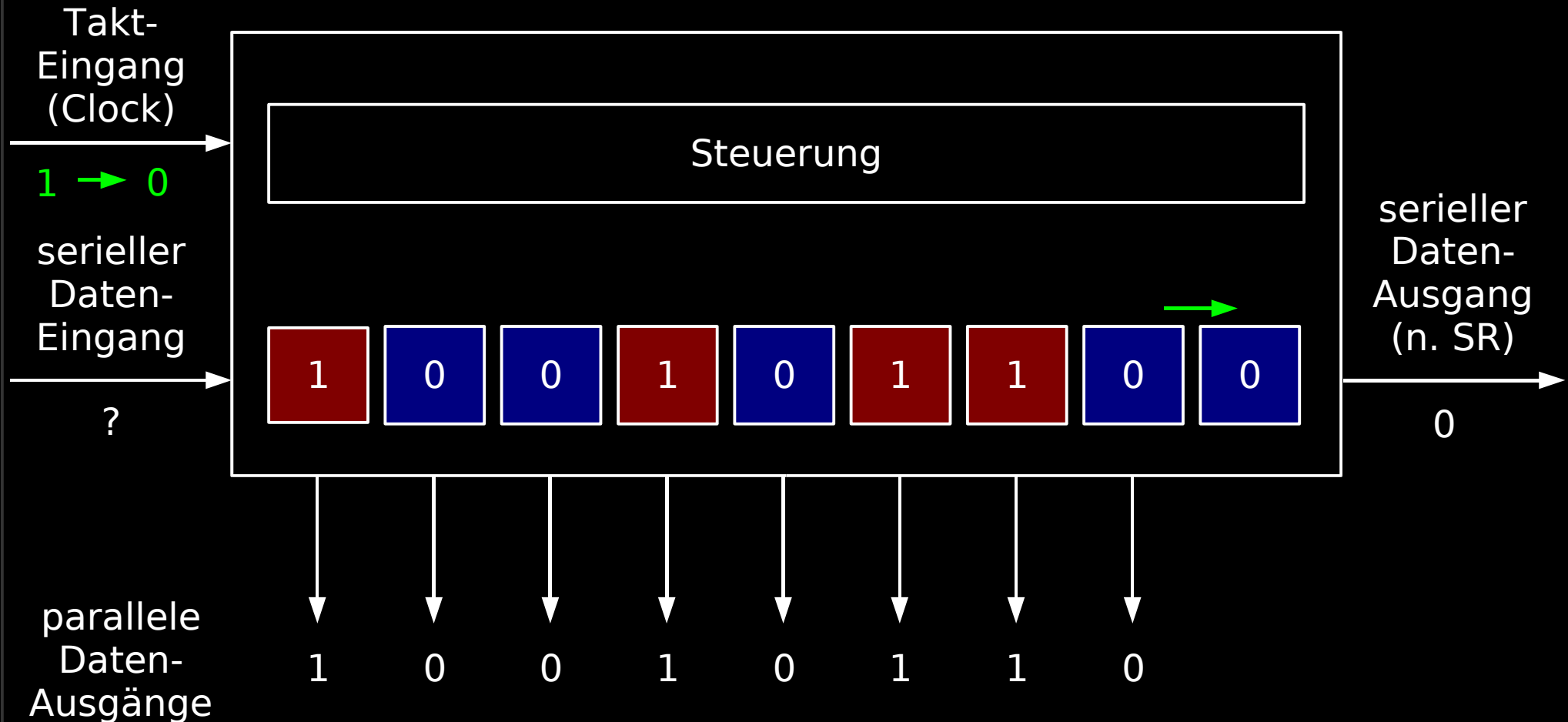
# Schieberegister – 7. Bit



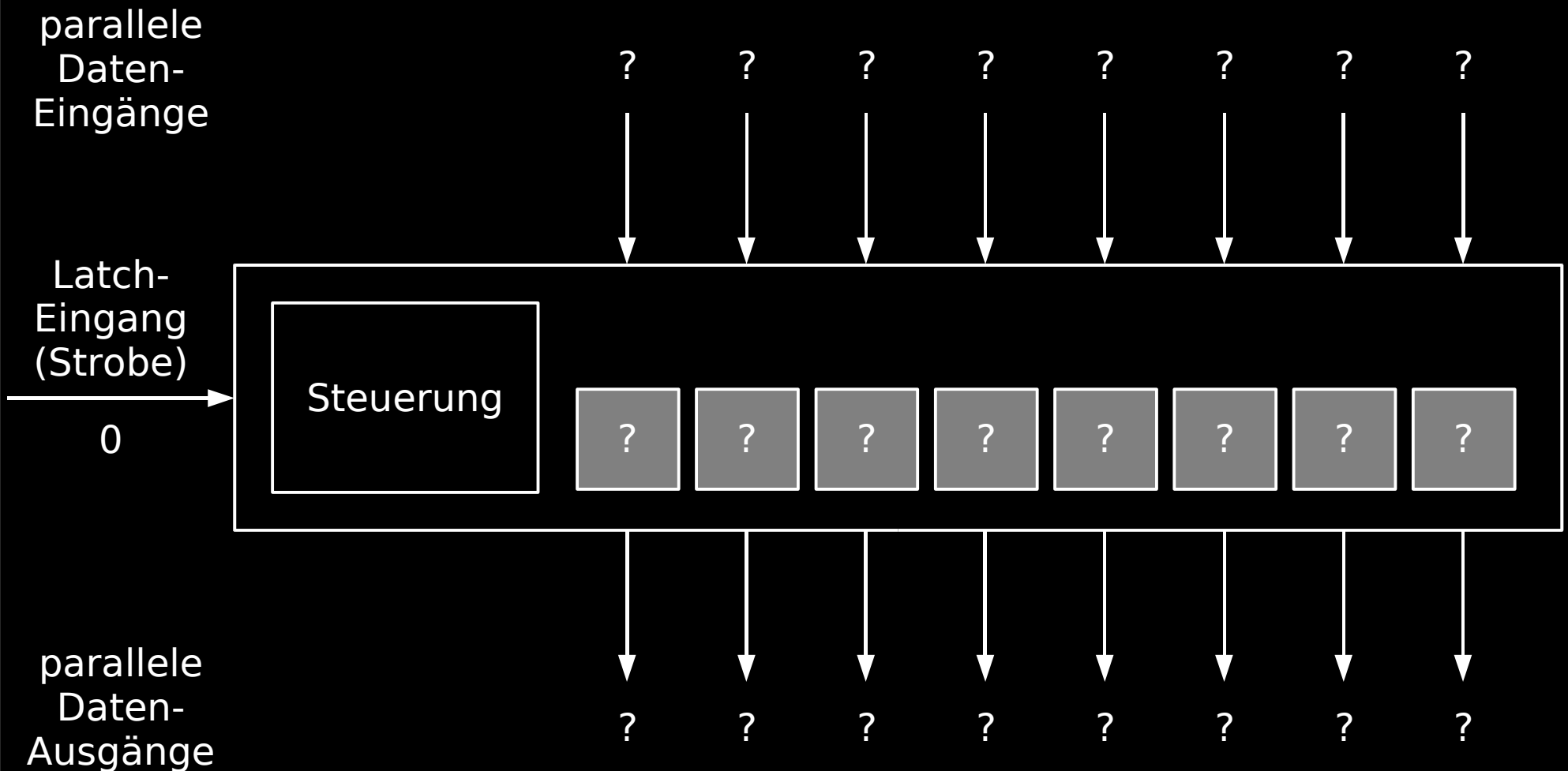
# Schieberegister – 8. pos. Flanke



# Schieberegister – 8. neg. Flanke

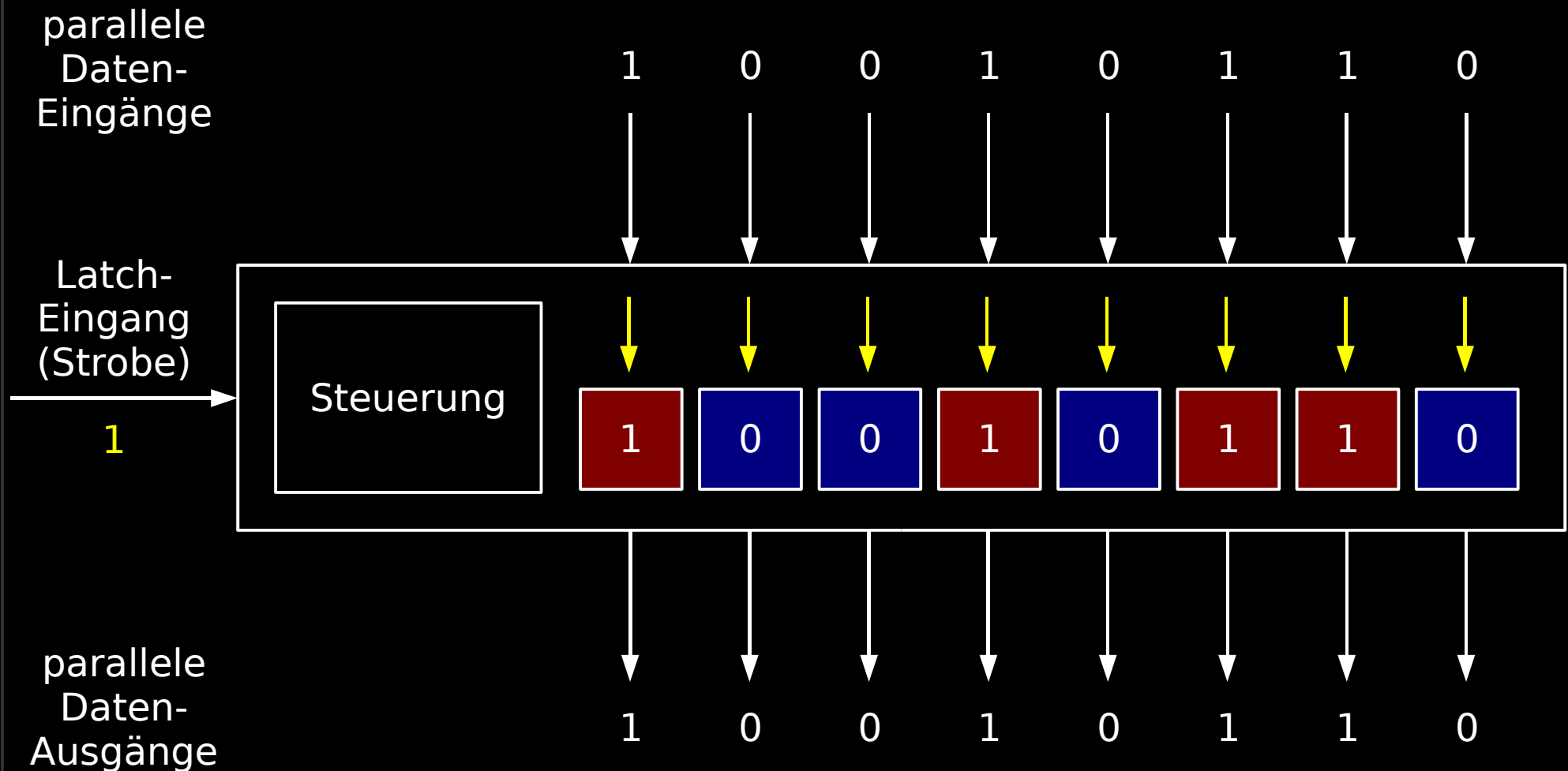


# Latch (Lt)

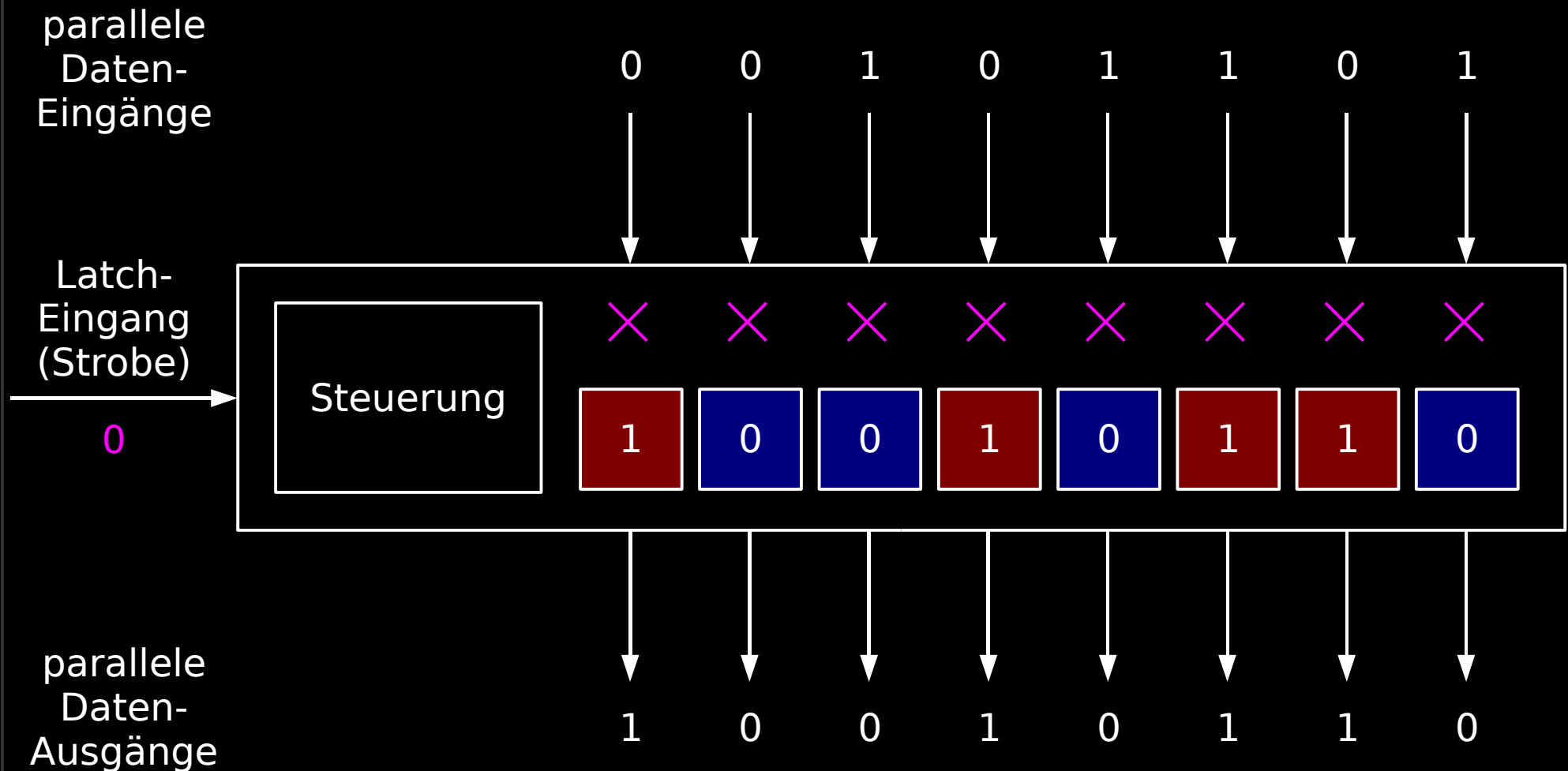




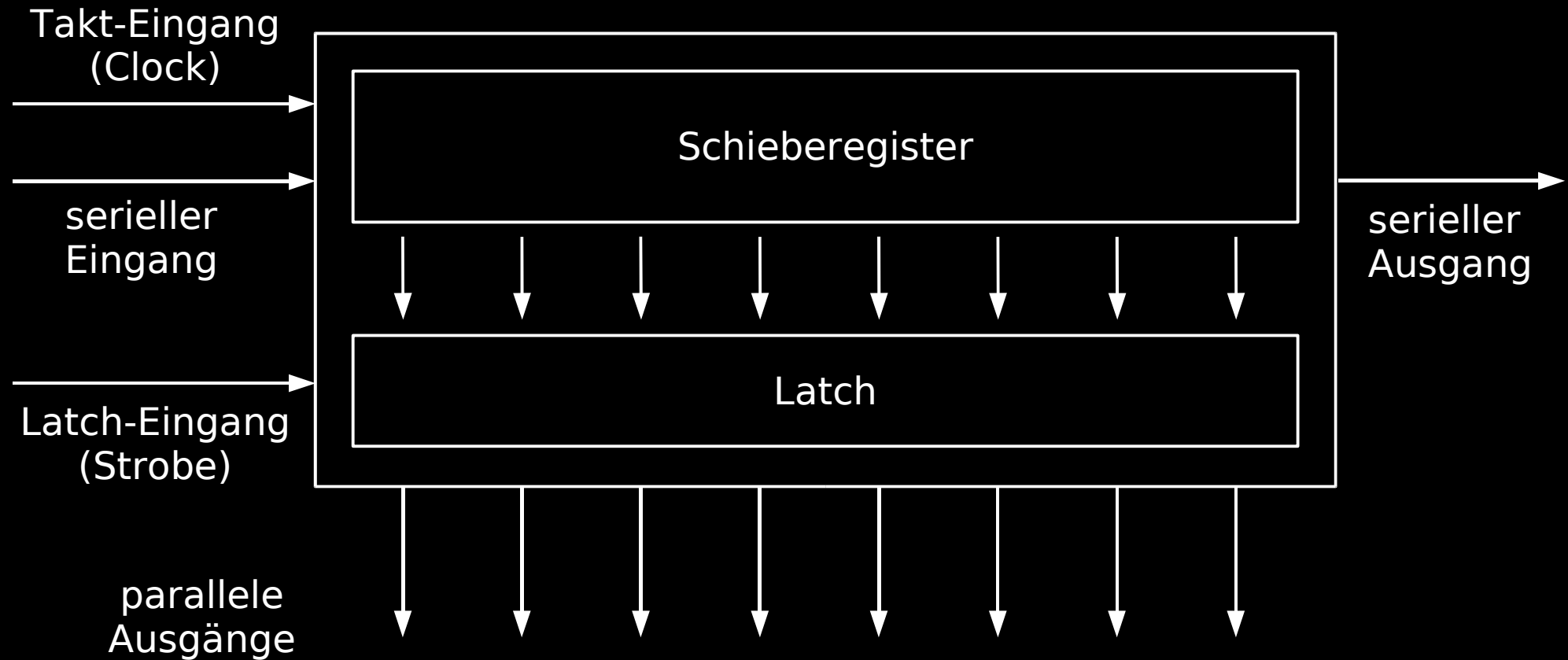
# Latch – Strobe aktiv



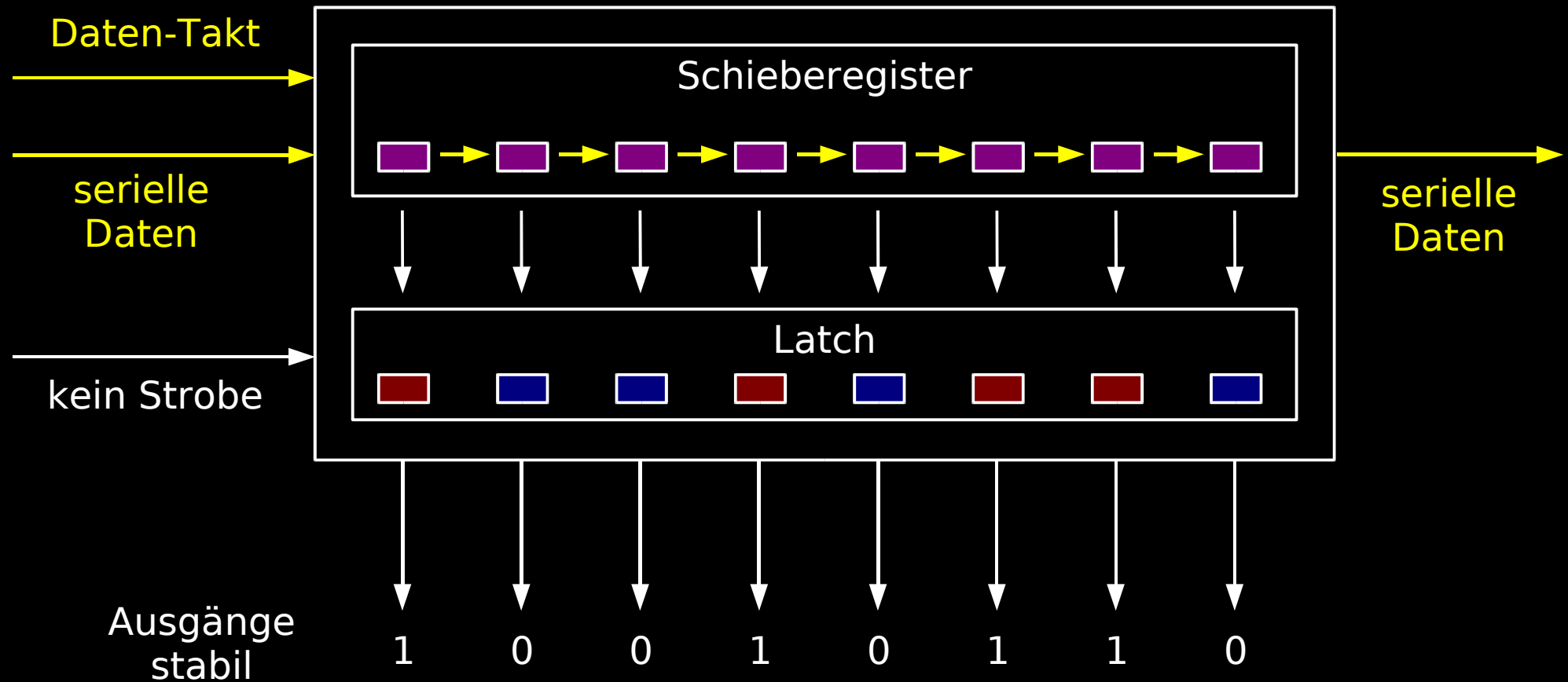
# Latch – Strobe inaktiv



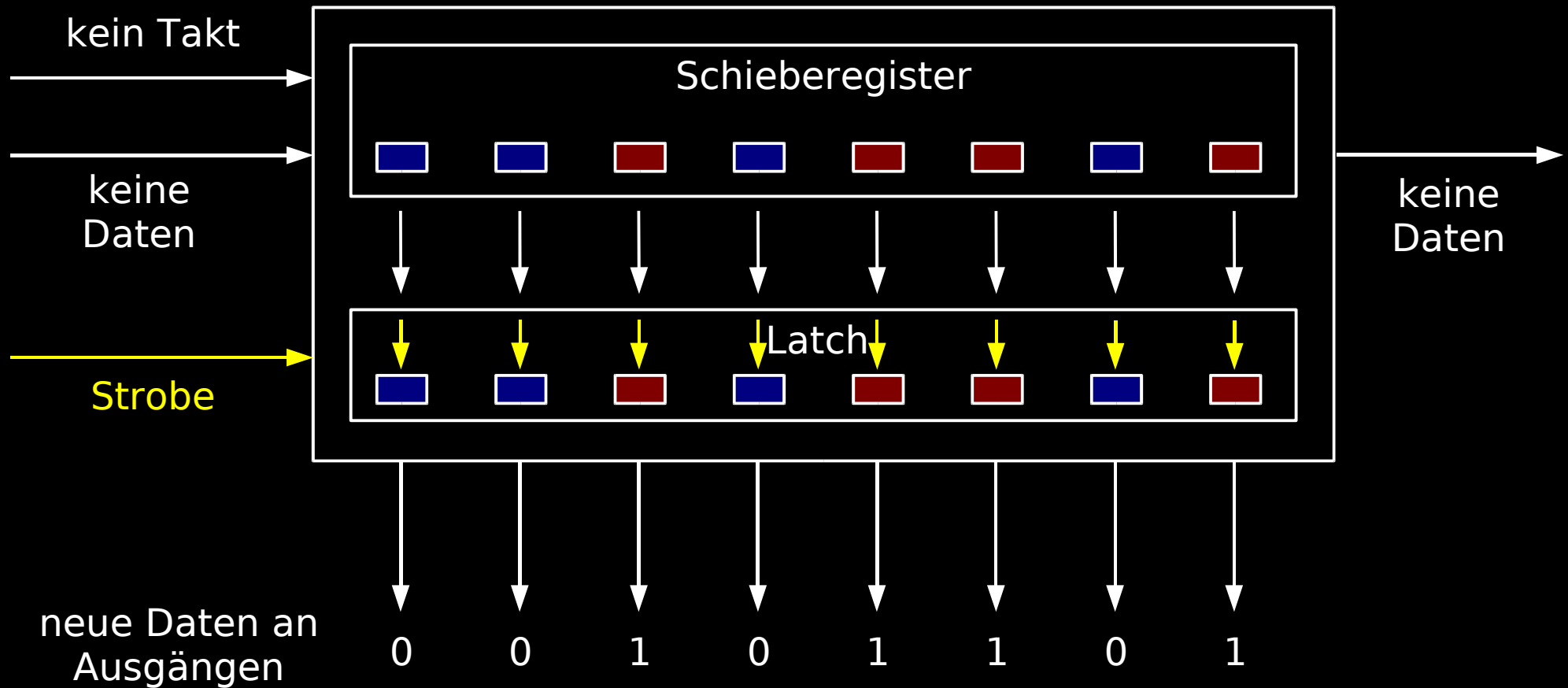
# 4094 / 74HC4094 (vereinfacht)



# 4094 – Daten laden

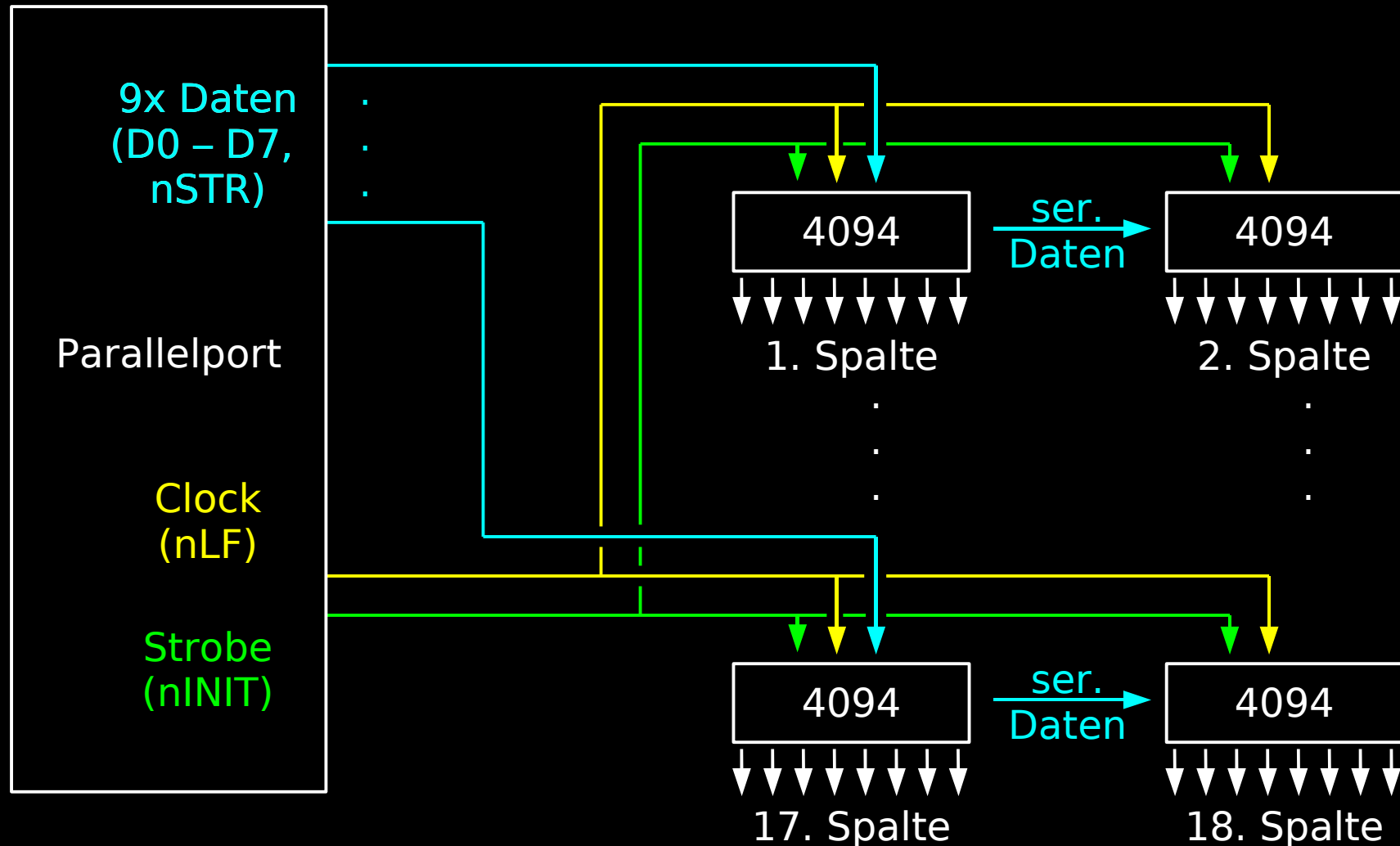


# 4094 – Daten übernehmen



# BlinkenLEDs-Schaltung

18x8 Pixel, schwarz/weiss



# BlinkenLEDs-basierte Projekte

- original BlinkenLEDs
- LittleLights V 1.0
  - Feldeffekt-Transistoren als Verstärker hinter 4094-Chips
  - Glühlampen
- BlinkenBoard
  - geätzte Leiterplatte
- BlinkenBlue
  - Verstärker hinter 4094-Chips
  - superhelle LEDs

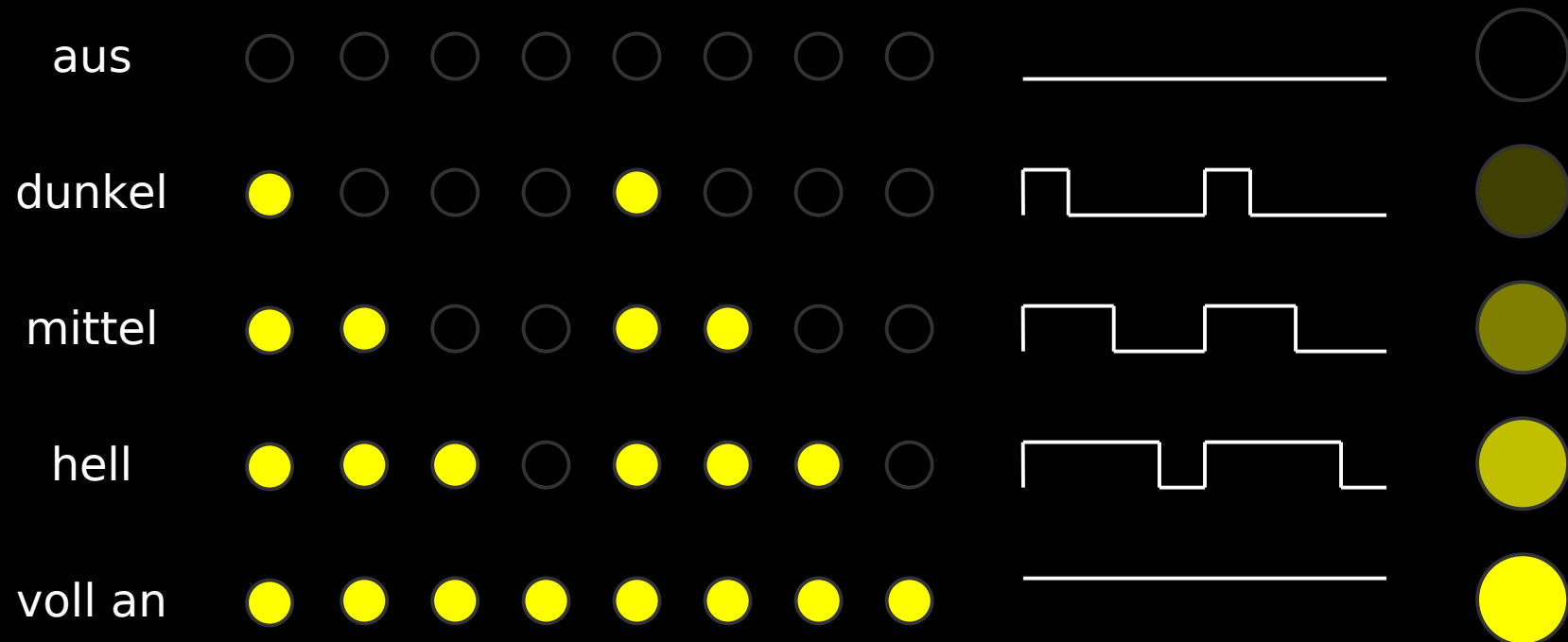
# BLINKENmini / BlinkenLEDsPro

- 18x8 Pixel
  - BlinkenLEDs-Schaltung
  - BLINKENmini: leicht andere Verdrahtung der 4094
- Graustufen
  - Pulsbreitenmodulation (PWM)
    - schnelles an-/aus-Schalten
      - ~2000 einzelne s/w-Bilder pro Sekunde
      - gesteuert von **Linux-Kernelmodul**



# Pulsbreitenmodulation (PWM)

Dimmen von Leuchtmitteln durch schnelles, unterschiedlich langes Einschalten



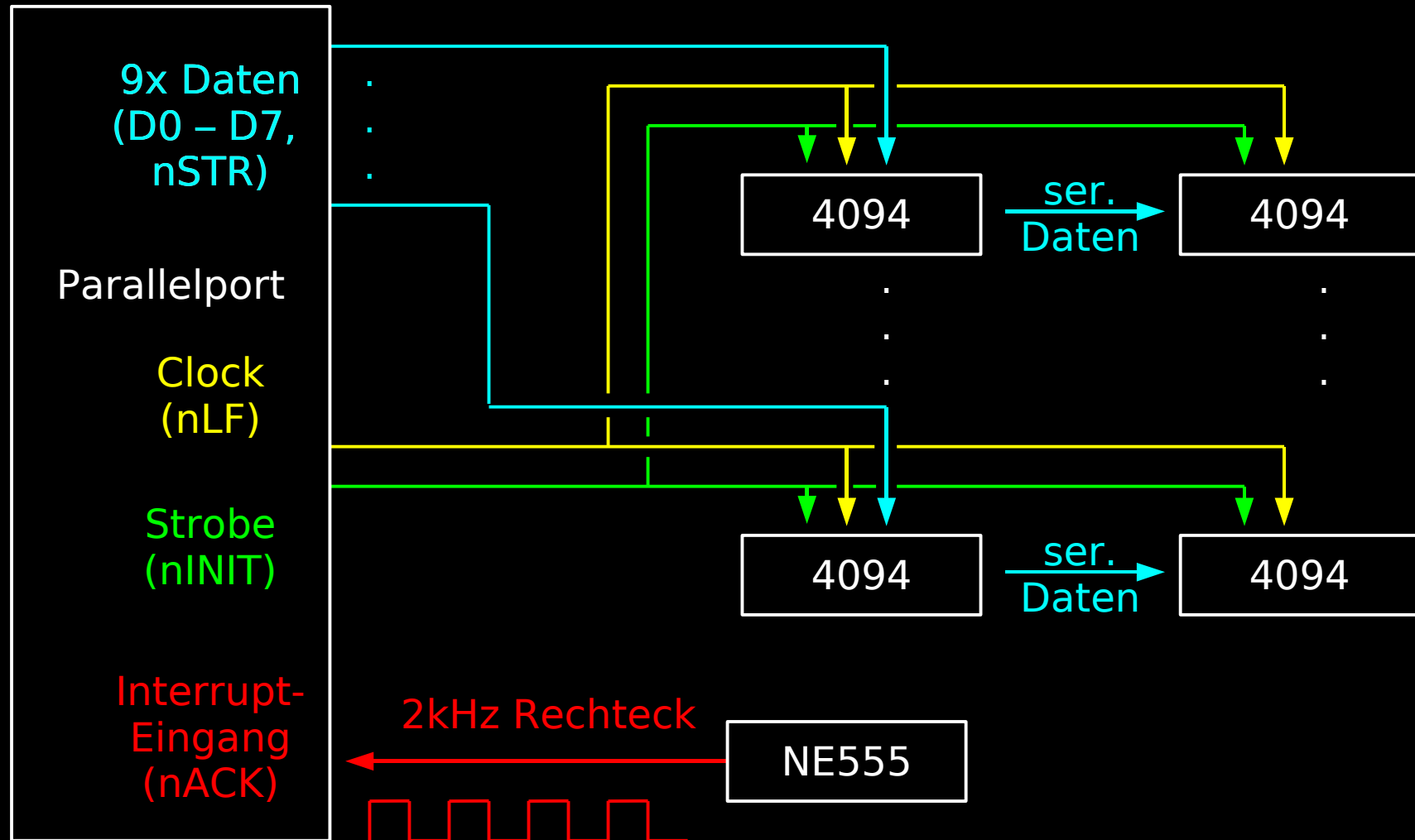
mind. 80 Zyklen pro Sekunde damit es nicht flackert

# BlinkenLEDsPro - Timing

- 16 Graustufen mit PWM
  - 15 s/w-Bilder pro Graustufen-Bild
- mind. 100Hz Bildwiederholfrequenz
  - mind. 1500 s/w-Bilder pro Sekunde
- PC-Betriebssysteme
  - Zeitscheibendauer: 1ms – 10ms
  - Reaktionszeit: einige Zeitscheiben
  - Bildausgabe mit 2000Hz nicht in Benutzer-Kontext möglich
  - Bildausgabe in Hardware-Interrupt

# BlinkenLEDsPro-Schaltung

18x8 Pixel, 16 Graustufen



# OS für BlinkenLEDsPro

## Anforderungen

- lauffähig auf normalen Desktop-PC
- Bildausgabe in Hardware-Interrupt-Handler
  - Entwicklung von Treibern einfach möglich
- Netzwerk-Unterstützung für Ankopplung an MCUF-Stream



**Linux** ist die Lösung



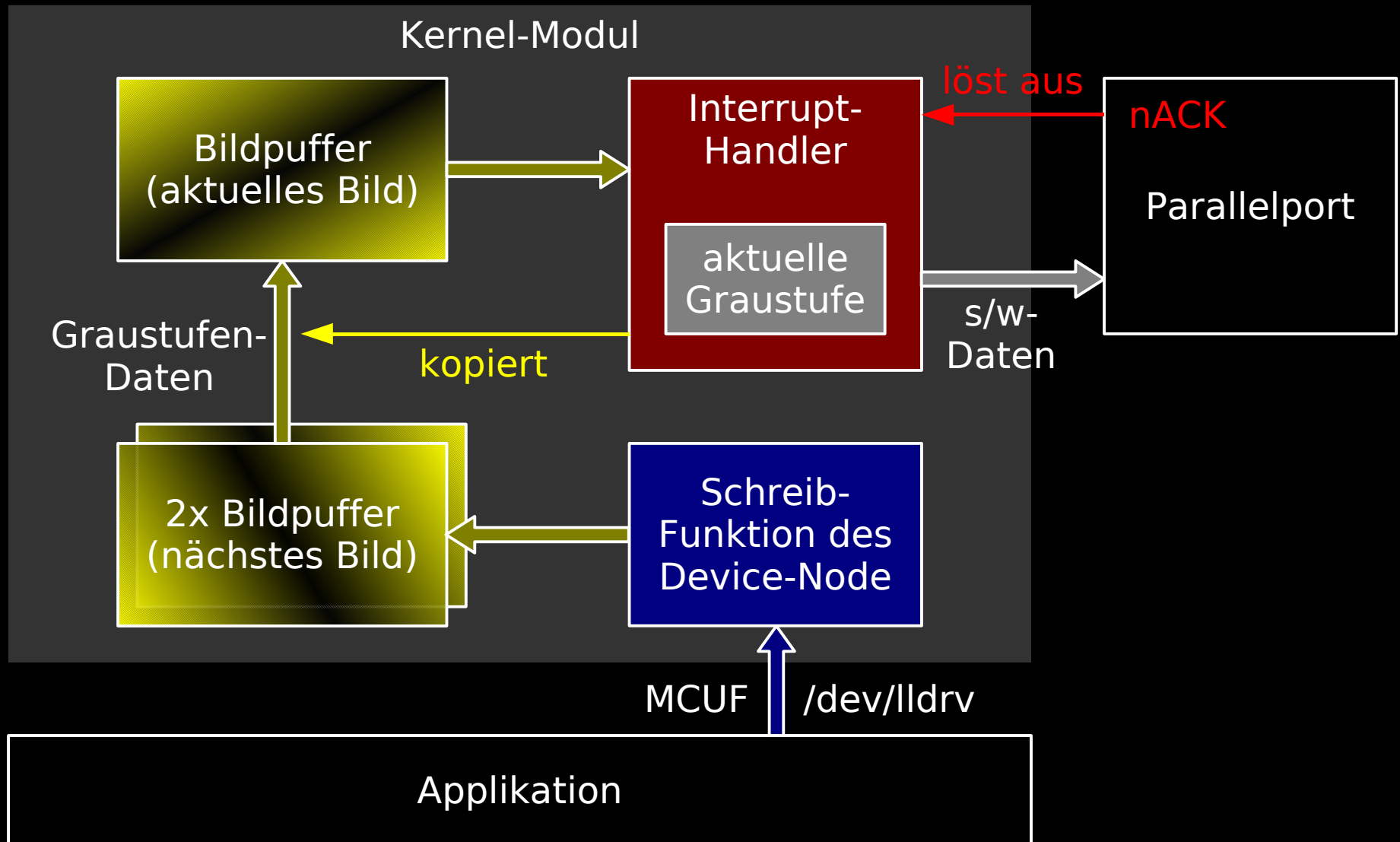
# LittleLights Driver - lldrv

Linux-Kernelmodul (für 2.4 und 2.6)

- zur Ansteuerung von
  - LittleLights V 2.0
  - BlinkenLEDsPro
  - BLINKENmini V 1.0
- Konfigurierbar
  - I/O-Adresse und Interrupt des Parallelports
  - Hardware-Typ (LittleLights / BLINKENmini)
- Eingabedaten
  - MCUF-Stream über Device-Node /dev/lldrv

<http://stefan.blinkenarea.org/lldrv/>

# Ildrv - Architektur



# Ildrv – Interrupt-Handler

2000 mal pro Sekunde

- für alle  $x=0..17, y=0..7$ 
  - $swBild[x,y] := aktBild[x,y] \geq aktGrau$
- `ausgeben( swBild )`
- `aktGrau++`
- wenn  $aktGrau \geq 16$ 
  - für alle  $x=0..17, y=0..7$ 
    - $aktBild[x,y] := nächstBild1[x,y]$
  - $aktGrau := 1$

# Ildrv - Schreib-Funktion

Schreibe MCUF-Frame auf Device-Node

- wenn MCUF-Header geschrieben
  - aktPos := Anfang
- wenn Daten geschrieben
  - naechstBild2[aktPos] := Daten
  - aktPos++
- wenn aktPos == Ende
  - vertauschen( naechstBild1, naechstBild2 )
  - aktPos := ungültig



# Ildrv – Portzugriffe

Portzugriffe für Ausgabe eines s/w-Bildes

- 9 Stränge mit 16 Bits / 2 Schieberegistern
- 4 Portzugriffe pro Bit
  - Daten: D0 – D7 ausgeben, nSTR ausgeben
  - Clock-Impuls: nLF setzen, nLF löschen
- 2 Portzugriffe für Strobe-Impuls
  - nINIT setzen, nINIT löschen
- **66 Portzugriffe** pro s/w-Bild
  - 16 \* 4 für Bits
  - 2 für Strobe

# Ildrv - Prozessorauslastung

- Parallelport ist ISA-Technik
  - Portzugriff mit 8MHz und I/O-Wait-States
  - Portzugriff  $\sim 1.5\mu\text{s}$ 
    - unabhängig von Prozessor- und Speichertakt
- Zeit für Ausgabe eines s/w Bildes
  - 66 Portzugriffe:  $\sim 100\mu\text{s}$
- Prozessorauslastung
  - $2\text{kHz} * 100\mu\text{s} = 20\%$  (rechnerisch)
  - durch Interrupt-Overhead:  $\sim 30\%$  (Messwert)

# Graustufen mit Microcontroller

## Problem

- PC-Hardware nicht optimal geeignet für PWM
- einfache, aber zeitkritische Operation
- viele Bit-basierte I/O-Operationen

## Idee

- **Microcontroller** erzeugt Graustufen
- Graustufen-Bilder vom PC
- s/w-Bilder an BlinkenLEDs

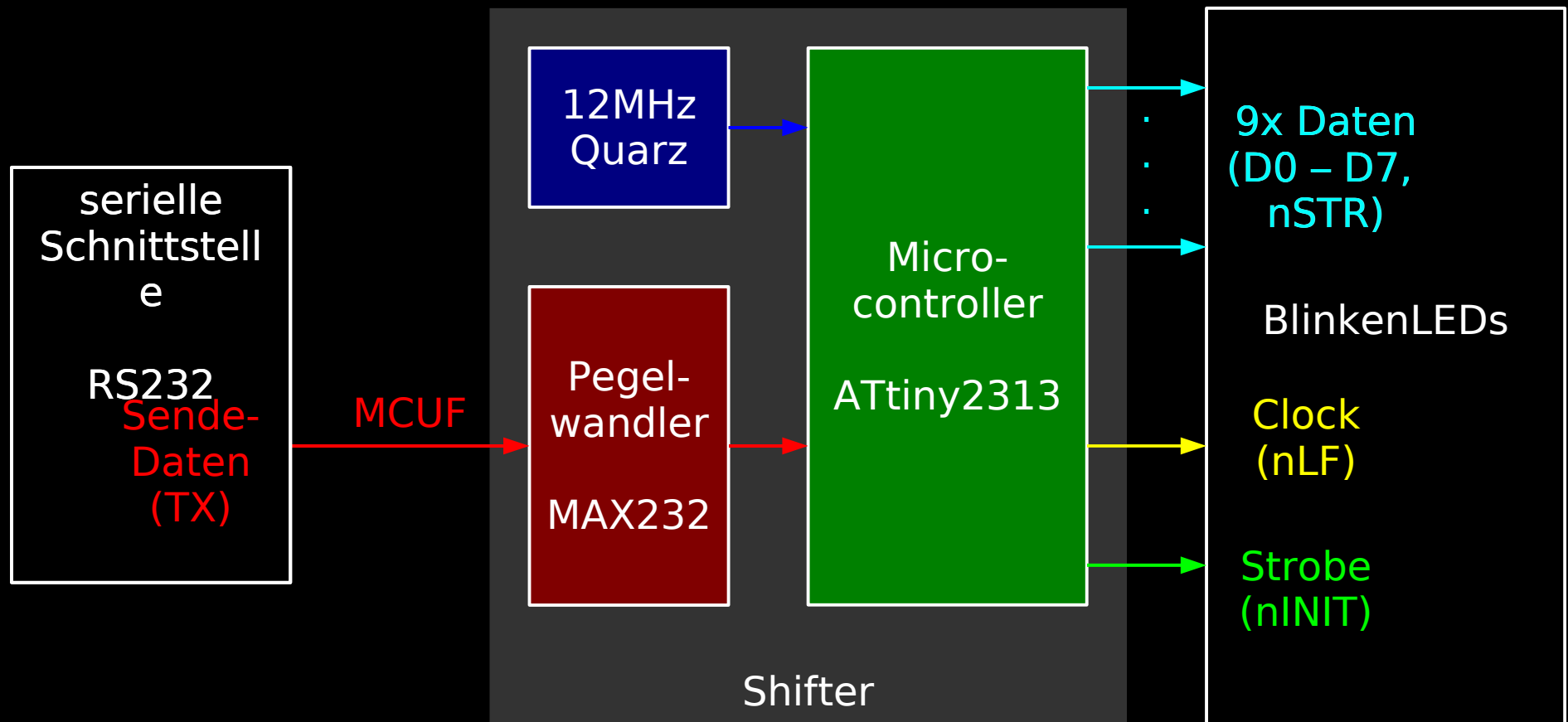
# Shifter

- Shifter als Add-On
  - zu BlinkenLEDs / LittleLights / BLINKENmini
- Microcontroller Attiny2313
  - Daten-Eingabe
    - MCUF, seriell, 8N1, 115200 bps
  - erzeugt 8 Graustufen mit Gamma-Korrektur
  - Ausgabe s/w-Bilder
    - über Datenleitungen, Clock und Strobe

<http://stefan.blinkenarea.org/Shifter/>

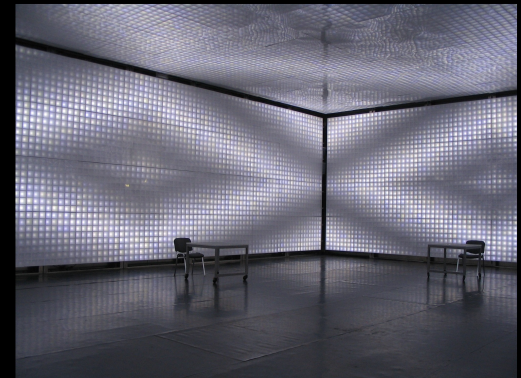
# Shifter-Schaltung

18x8 Pixel, 8 Graustufen mit Gamma-Korrektur

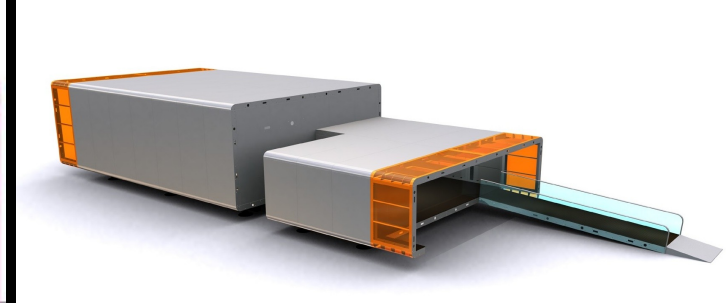
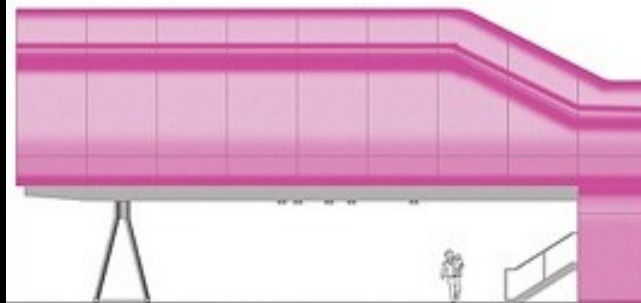


# Eigenständige Projekte: TROIA

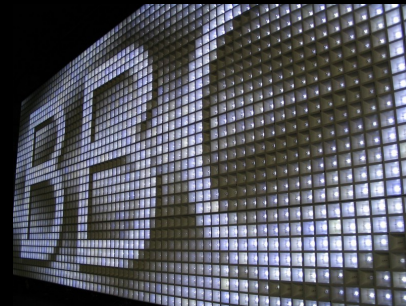
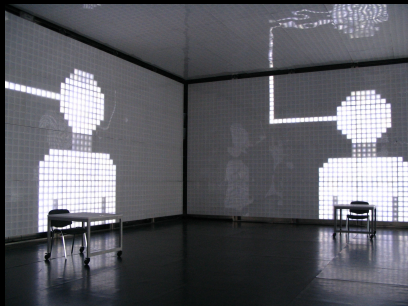
- Temporary Residence of Intelligent Agents
- Zusammenarbeit: BBM / BBO
- Zeitraum: 2004 / 2005
- Thema: Technologien der politischen Kontrolle
- Kulturprojekt
  - Künstler, Architekten, Theatermacher, Wissenschaftler, Programmierer und Ingenieure aus sieben europäischen Ländern
  - Multimedia Theaterstück
  - experimentelle Plattform: intelligenter Raum + mobiles Gebäude



# Eigenständige Projekte: TROIA – Gebäude / Raum

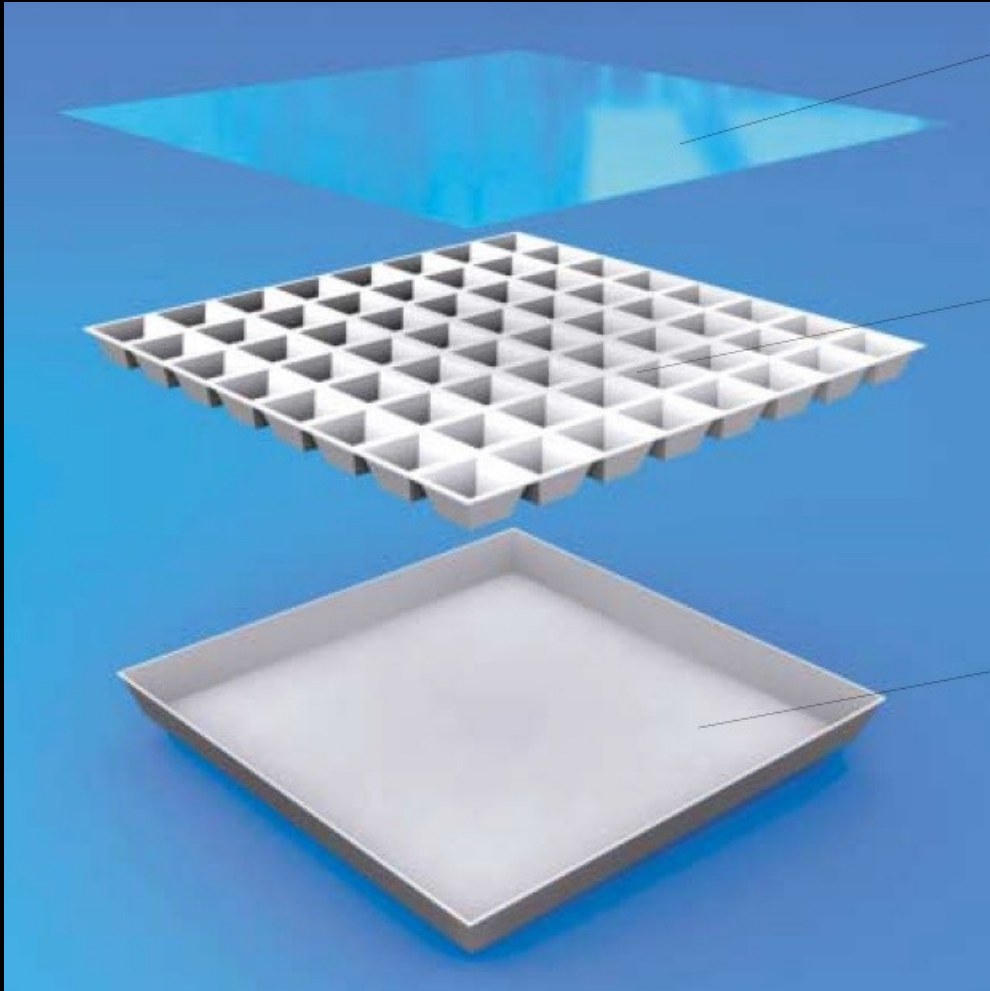


- Gebäude: ca. 20 m x 12 m x 6 m
- Pixelraum: 14,56 m lang, 11,20 m breit und 4,48 m hoch
  - 20096 Pixel mit 128 Graustufen
  - Platz für ca. 200 Personen
  - Grundfläche: 163 m<sup>2</sup>
  - Pixelfläche: 400 m<sup>2</sup>





# Eigenständige Projekte: TROIA - Pixelkisten

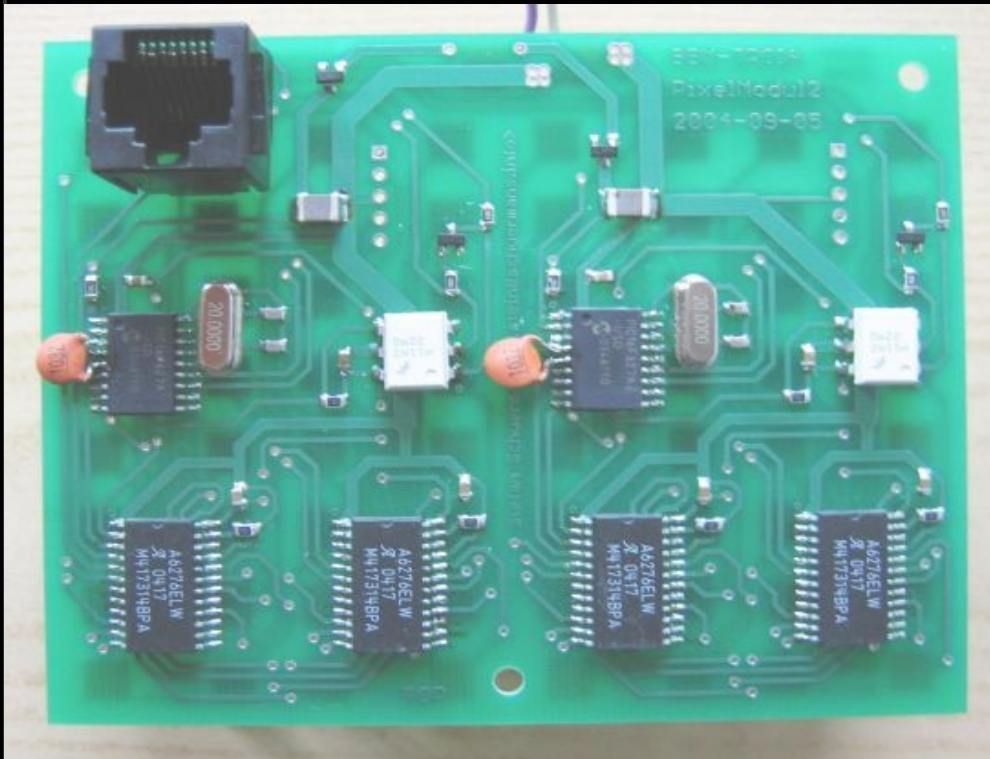


- 8x8 Pixel (1,12 m x 1,12 m)
- Ansteuerungselektronik:
  - Pixelmodulplatine
  - interne Verkabelung
- 2 Stecker:
  - XLR-Einbaustecker für die Betriebsspannung 5 V mit 1.5 A (100 mA fuer ein Pixelmodul reicht)
  - RJ45-Buchse für die Daten



# Eigenständige Projekte: TROIA – Ansteuerung

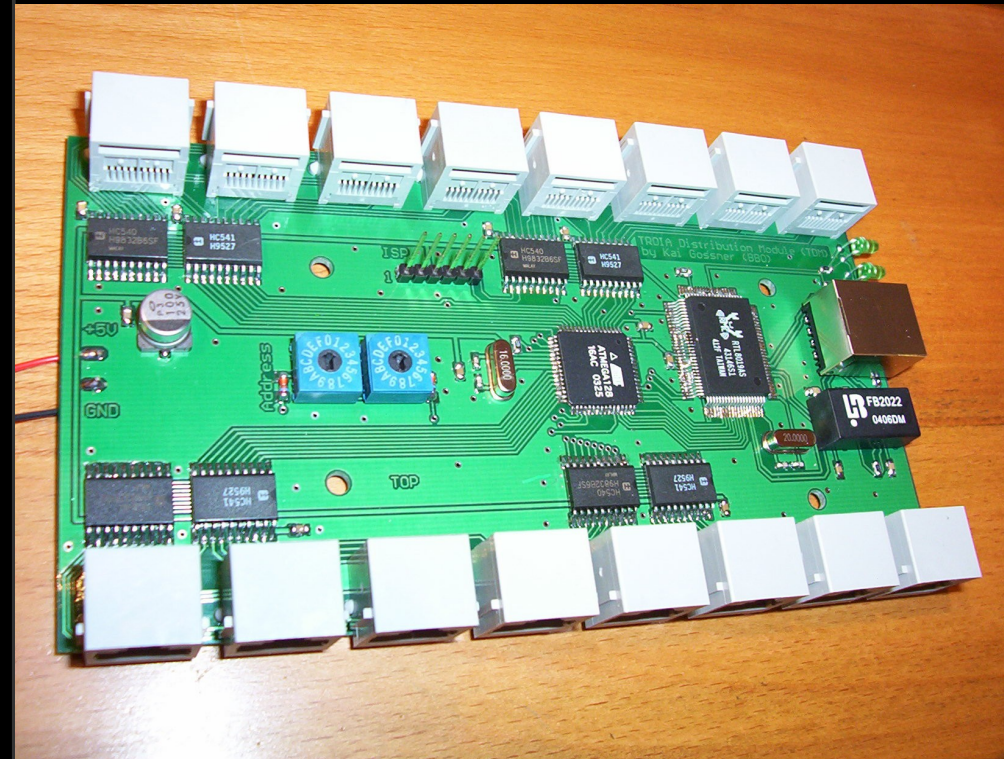
## Pixelmodul



### Aufgaben:

- seriellen Datenstrom entgegennehmen
- PWM-Signal für 32 Pixel generieren

## Verteilermodul



### Aufgaben:

- bis zu 1024 Pixel ansteuern
- ankommenden Daten in 32 Ströme serieller Daten aufteilen

Film

# Eigenständige Projekte: bluebox – Geschichte

2004:

- Plattenbau in der Magdeburger Innenstadt
- „Blauer Bock“
- statischer Adventskalender
- gemeinnütziger Verein aus Magdeburg
- einfache Installation mit Leuchtstofflampen & mechanischen Zeitschaltuhren



2006:

- Fassadendisplay bluebox
- BlinkenArea als Partner für technische Umsetzung
- computergesteuerte Installation zur Ansteuerung jedes einzelnen Fensters als Pixel

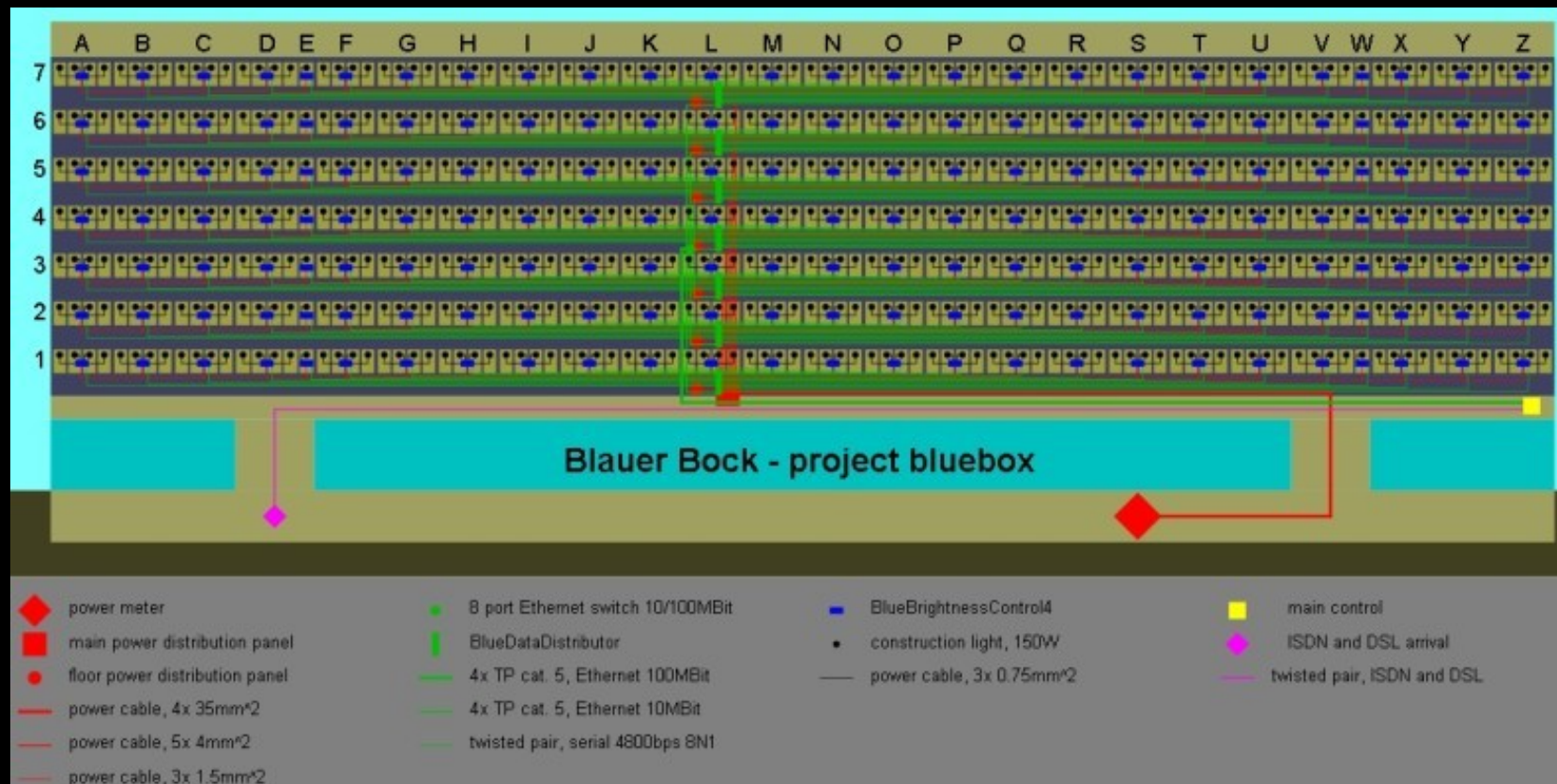


# Eigenständige Projekte: bluebox – Setup I



- Räume
- Energieverteilung
- Datenverteilung

[http://bluebox.blinkenarea.org/tech\\_doc/index.html](http://bluebox.blinkenarea.org/tech_doc/index.html)



# Eigenständige Projekte: bluebox – Setup II

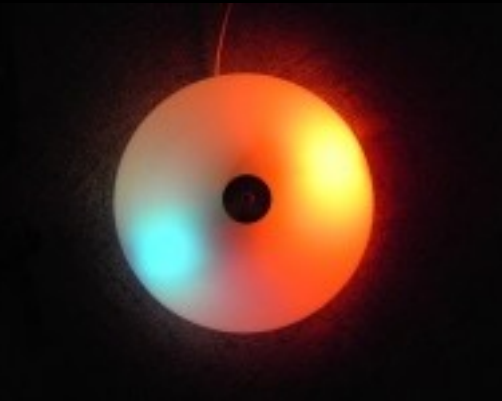
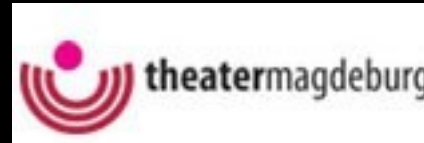
## Material

- 686 Baustrahler + 100 Ersatzbrenner
- 1000 m<sup>2</sup> Dampf- und Windsperre
- 200 Dachlatten
- ca. 5 km Datenkabel
- ca. 2 km Stromkabel
- 192 Steuerungsplatinen



# Eigenständige Projekte: bluebox – Umfeld

- Filmwettbewerb
- Jugendwettbewerb „freistil“
- nachtcafé bluebox
- bluebox Glasbox
- bluebox Postkarte
- Heiratsantrag am Valentinstag
- Spielstände der Handball-WM



# Eigenständige Projekte: bluebox

Film



# Stand der Dinge: Allgemein

- aktuelle Informationen im Blog
- Dokumentation aller Projekte / Aktionen / Events im Wiki
- Diskussionen und Support im Forum
- Mailingliste offen für Interessierte
- wöchentliches Meeting im IRC
- viele verschiedene Menschen und Projekte





# Stand der Dinge: Spendenaktionen

- Bausätze – Spende an Unicef
- Spendenaktion bluebox
- Spendenaktion BlinkenSisters



# Stand der Dinge: Ausstellungen

- LittleLights
- pong.mythos



- City-Hochhaus Leipzig
- public play



# Ausblick

## Events

- Easterhegg 2008 in Köln
- ICMP4 in Münchsteinach
- Chaos Communication Congress 2008 (25C3) in Berlin

## Projekte

- blueboxMini
- Bewerbung für RUHR.2010

## Weitere Informationen

- im Wiki: <http://wiki.blinkenarea.org>
- im Blog: <http://blog.blinkenarea.org>

## Neue Projekte?



# Vielen Dank!

```
The system is going down for system halt NOW!  
INIT: Switching to runlevel: 0  
INIT: Sending processes the TERM signal
```