

The Blinking Suspects

21C3 | 27. 12. 2004



v.1.01

Stefan '1stein' Schürmans
& Stephan 'ST' Kambor



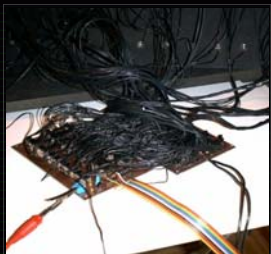
Abl auf:



Hardware Projekte mit 144 Pixeln
Dateiformate
Hardware Projekte mit 520 Pixeln
Netzwerkprotokolle
Software Projekte
Zukunft?



Ende



30 Minuten



<https://blinkenarea.org/>

copyleft - all rights reversed

The Blinking Suspects

21C3 | 27. 12. 2004

HW mit 144 Pixeln

Wie steuere ich 144 Pixel (preiswert) mit einem Rechner an?

Problem:

Jede Lampe benötigt zu einem bestimmten Zeitpunkt ein definiertes Signal [0/1]
144 Lampen benötigen 144 Ausgänge am PC
Parallele Schnittstelle -> 25 Pins (nicht alle können als Ausgang genutzt werden)

Lösung:

1. 3 ISA Karten mit je 48 Ausgängen -> hohe Kosten
2. MCUs und Ethernet -> hohe Kosten
3. irgend eine andere Lösung ...
4. mit 8 Bit Schieberegistern



<https://blinkenarea.org/>

copyleft - all rights reversed

Wie sieht ein 8 Bit Schieberegister aus?

IC - Integrated Circuit
16 Pins



Wie funktioniert ein 8 Bit Schieberegister?

mehrere miteinander gekoppelte Speicherzellen

Dateneingang -> Bitkombination [10100101]

Takt

8 Ausgänge

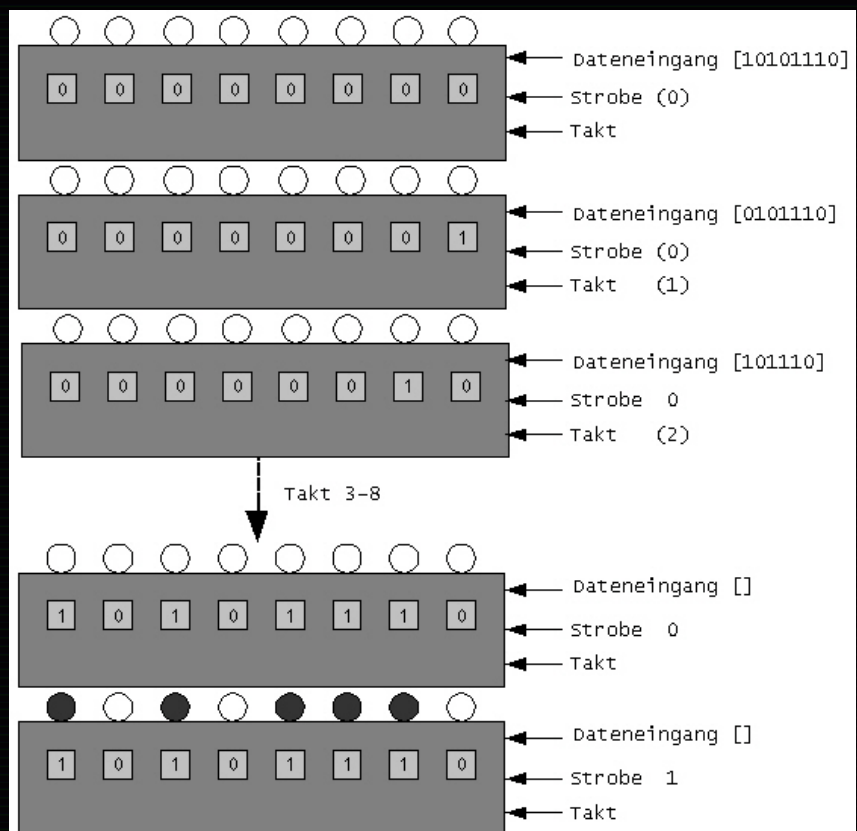
Daten werden bei jedem Takt von Speicherzelle zu Speicherzelle weitergeschoben

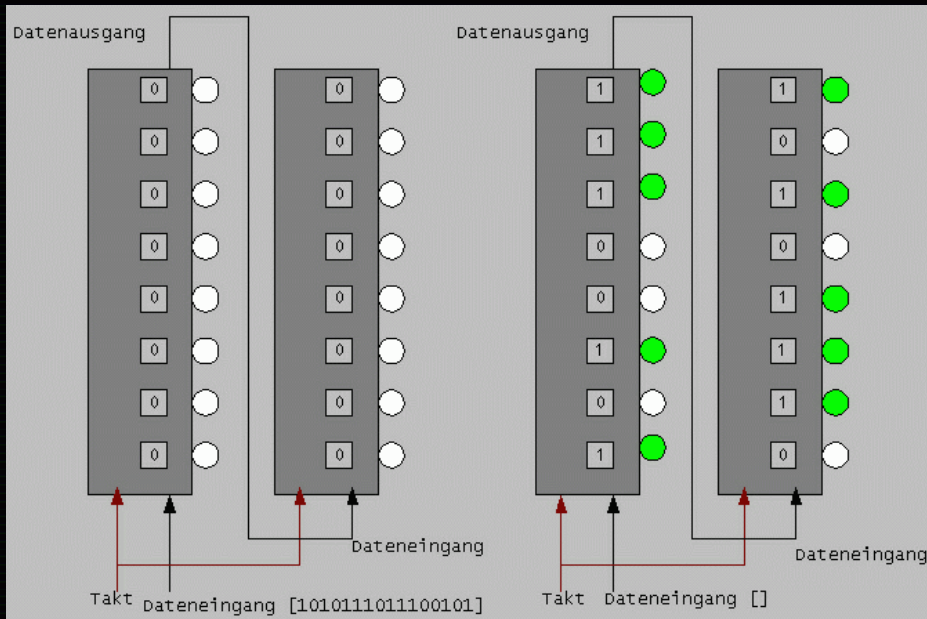
Beispiel 8 Bit Schieberegister/Latch:

Bitkombination wird bei jedem Takt um ein Bit in das Register geschoben

Nach 8 Takten "Strobe", die Bitkombination wird ausgegeben

8 "Lampen" können mit einem Ausgang des Parallelports gesteuert werden -> 18 Ausgänge -> immernoch zuviele





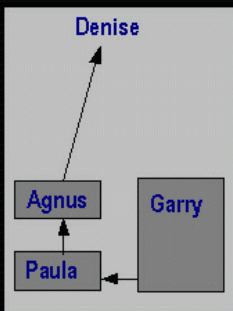
Register haben auch Ausgänge
~> kaskadieren möglich

16 Lampen ~> 2 Spalten ~> 1 Ausgang

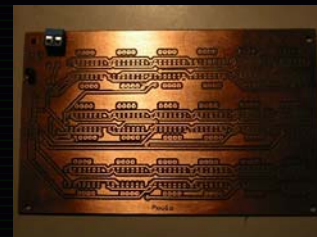
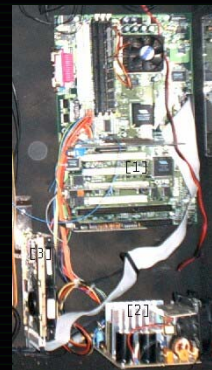
9 Ausgänge werden benötigt

LittleLights

Hardware / genereller Aufbau



Garry ~> Rechner
Paula ~> Steuerplatine
Agnus ~> Verstärkerplatine
Denise ~> Haus [Gehäuse]



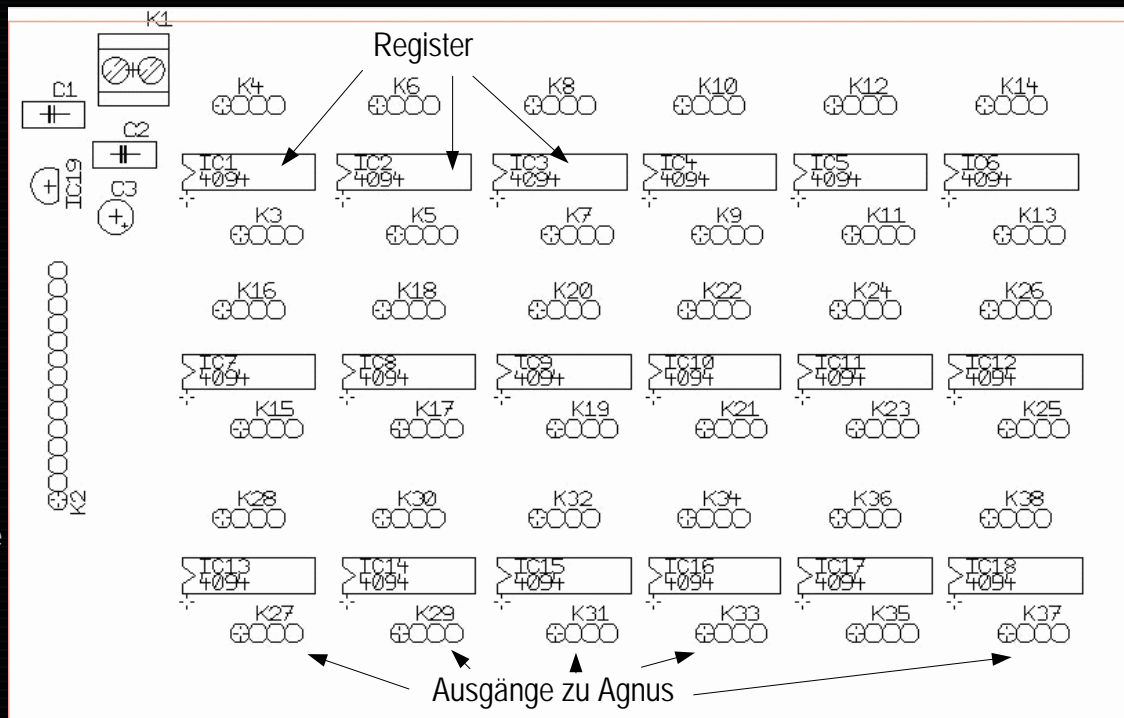
Hardware / Paula

18 Register

Daten von Garry

- 9 x Daten
- 1 x Masse
- 1 x Strobe
- 1 x Clock
- 1 x Output Enable

144 Signale zu Agnus

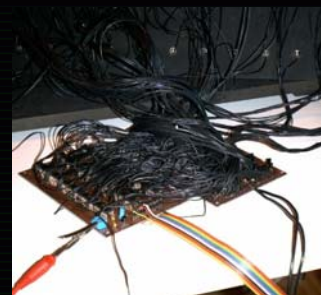
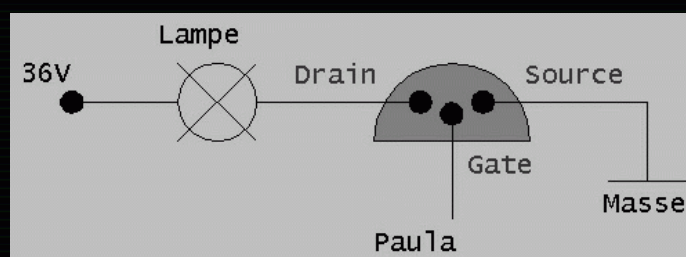


Hardware / Agnus

Paula ~> 5V
 Registerausgänge nicht belastbar
 Lampen ~ 36 V
 Verstärkung notwendig



144 N-Kanal Anreicherungs MOS-FETs in low-side-Verstärkerschaltung



Hardware / Garry



Standard PC:

- [1] Mainboard
[Prozessor: AMD K6 II 450 MHz / 256 MB RAM]
- [2] Netzteil
[230 Watt ATX]
- [3] Festplatte
[IDE 545,5 MB]

Garry bereitet die Daten für Paula auf

Netzwerkschnittstelle / UDP

Prozessor untertaktet auf 400 MHz ~> geringere Wärmeentwicklung

BlinkenLEDs



<http://www.blinkenleds.de/>
<http://www.jalcdsforum.de/>

BlinkenBoard - <http://blinkenleds.spline.de/>



BlinkenBlue 1 +2 - <http://www.moddingmeister.de/>



BlinkenLEDs Pro - <https://wiki.blinkenarea.org/>



und noch viele mehr...

XMasLi ghts

The Blinking Suspects
21C3 | 27. 12. 2004



288 Lichterkettenbirnchen
288 BC337 NPN-Transistoren
288 Widerstände
20 74164 TTL-Schieberegister
At90s2313
Max232
viel Kabel und Lötzinn

Ansteuerung über serielle Schnittstelle (9600Baud 8N1)

<https://wiki.blinkenarea.org/bin/view/Blinkenarea/XMasLights>



<https://blinkenarea.org/>

copyleft - all rights reversed

BLINKENmini - Graustufen

The Blinking Suspects
21C3 | 27.12.2004

Taktgenerator



Parallelpor

ACK (Pin 1

<https://blinkenarea.org/>

copyleft - all rig

Eine *.blm Datei besteht aus einem Header und den Nutzdaten die angeben, zu welchem Zeitpunkt welche Lampe leuchtet.

Die Headerdaten erklären sich soweit von selbst (Beispiel: Littlelights Intro):

```
# BlinkenLights Movie 18x8 // Dateiformat
# name = LittleLights // Name des Films
# description = LittleLights Intro // zusätzliche Beschreibung
# creator = Blinkenpaint 2.4 // Programm mit dem es erstellt wurde
# author = ST // Author
# width = 18 // Breite in Pixeln
# height = 8 // Höhe in Pixeln
# loop = no // Wiederholung
# duration = 27450 // Dauer in ms
```

Die Nutzdaten bestehen aus mehreren Bildern, wobei eine 1 für eine eingeschaltene Lampe, eine 0 für eine ausgeschaltene Lampe steht. Die Zahl nach dem @ bestimmt, wieviel ms das Bild dargestellt werden soll.

```
@500 @100 @1000
00000000000000000000 00000000000000000000 011111111000000000
000000011100000000 000000100100100000 100010100100000000
000000100010000000 000000010101000000 100001111111111100
000000000110000000 000000001110000000 100001111111101110
000000001100000000 000111111111111000 100010100100011100
000000000110000000 000000001110000000 011111111000000100
000000100010000000 000000010101000000 000000000000011010
000000011100000000 000000100100100000 000000000000101001
```

Eine *.bmm Datei besteht aus einem Header und den Nutzdaten die angeben, zu welchem Zeitpunkt welche Lampe wie hell leuchtet.

Die Headerdaten erklären sich soweit von selbst:

```
# BlinkenMini Movie 18x8 // Dateiformat
# name = Test // Name des Films
# creator = vi // Programm mit dem es erstellt wurde
# author = 1stein // Author
# width = 18 // Breite in Pixeln
# height = 8 // Höhe in Pixeln
# loop = no // Wiederholung
# duration = 27450 // Dauer in ms
```

Die Nutzdaten bestehen aus mehreren Bildern, wobei 0xFF für eine voll eingeschaltene Lampe, 0x00 für eine ausgeschaltene Lampe steht. Die Zahl nach dem @ bestimmt, wieviel ms das Bild dargestellt werden soll.

```
@800
0x00 0x00 0x00 0x00 0xFF 0xFF 0xFF 0x00 0x00 0x00 0xFF 0xFF 0xFF 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0xFF 0x7F 0x7F 0x7F 0xFF 0x00 0xFF 0x7F 0x7F 0x7F 0xFF 0x00 0x00 0x00 0x00
0x00 0x00 0xFF 0x7F 0x7F 0x7F 0x7F 0xFF 0x7F 0x7F 0x7F 0x7F 0x7F 0xFF 0x00 0x00 0x00
0x00 0x00 0x00 0xFF 0x7F 0x7F 0x7F 0x7F 0x7F 0x7F 0x7F 0x7F 0x7F 0xFF 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0xFF 0xFF 0x7F 0x7F 0x7F 0x7F 0x7F 0x7F 0xFF 0xFF 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0xFF 0x7F 0xFF 0xFF 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

```
<?xml version="1.0" encoding="UTF-8"?>
<blm width="18" height="8" bits="4" channels="1">
  <header>
    <title>bbo-bbm</title>
    <description>Auto-Generated Vertical Color Test</description>
    <creator>Perlsript</creator>
    <author>CAVAC</author>
    <email>cavac@grumpfzotz.org</email>
    <loop>no</loop>
  </header>

  <frame duration="50">
    <row>89ABCDEFEDCBA98765</row>
    <row>789ABCDEFEDCBA9876</row>
    <row>6789ABCDEFEDCBA987</row>
    <row>56789ABCDEFEDCBA98</row>
    <row>456789ABCDEFEDCBA9</row>
    <row>3456789ABCDEFEDCBA</row>
    <row>23456789ABCDEFEDCB</row>
    <row>123456789ABCDEFEDC</row>
  </frame>
  ...
  ...
  ...
</blm>
```

ähnlich blm nur im XML Format

pro Pixel sind nicht nur 2 Zustände möglich
links 16 [hex 0 - F]

Binärformat
kleiner
schneller

*.bmm Datei:
main header
[zusätzliche header]
frame start marker
frames

main header:
magic (32 bit)
height (16 bit)
width (16 bit)
channels (16 bit)
maxval (16 bit)
framecnt (32 bit)
duration (32 bit)
frameptr (32 bit)

Größenvergleich

```
-rw-r--r-- 1 1stein users 1714 Dec 23 11:58 test.blm
-rw-r--r-- 1 1stein users 4309 Dec 23 11:58 test.bml
-rw-r--r-- 1 1stein users 7892 Dec 23 11:58 test.bmm
-rw-r--r-- 1 1stein users 1648 Dec 23 11:58 test.bmm
```

frame start marker:
magic (32 bit) ('f' 'r' 'm' 's')

frame:
duration (16 bit)
pixels (je 8 bit)

Ausschnitt:

```
1d0 40 30 30 30 20 10 00 00 00 00 00 00 00 00 00
1e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*
250 00 00 00 00 00 00 00 00 00 00 00 00 64 00 00
260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

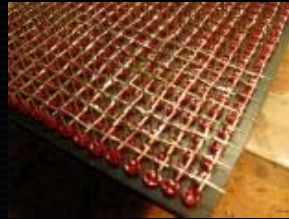

ARCADE mini

RotArcade

[<https://www.arcademinis.chuermans.info/>]



- Matrix aus 520 oder 1040 LEDs



- 16 Rot-Stufen (PWM)
- standalone
- programmierbar
- Compact-Flash
- USB
- 2 SNES-Pads



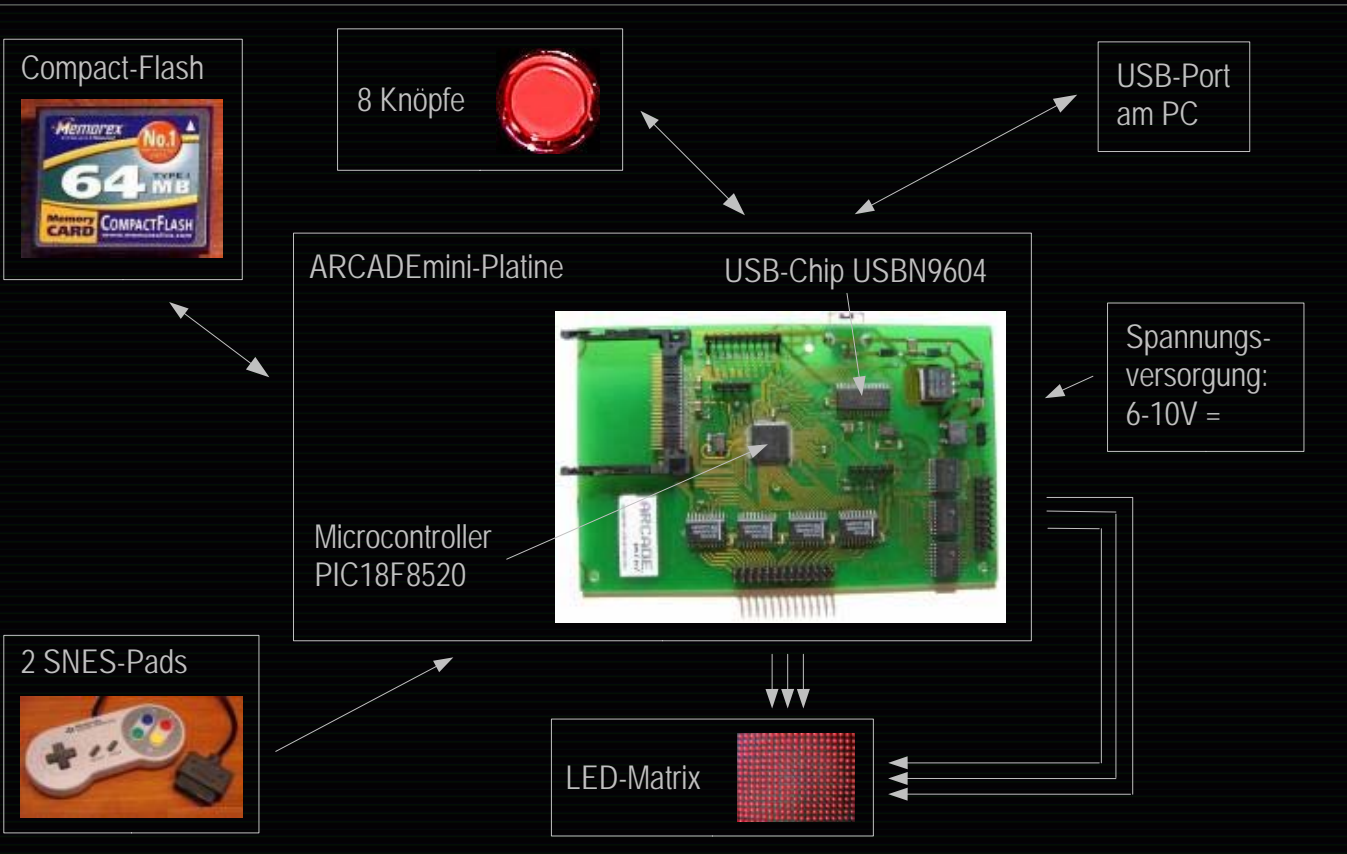
ARCADEmini gibt es teilweise als Bausatz



[<https://wiki.ccc.de/bin/view/Camp2003/RotArcade>]

- Propeller-Clock-Prinzip:
 - 20 LEDs rotieren mit 3000 U/min
 - 8 Rot-Stufen (PWM)
 - 512 x 20 Pixel
 - Filme im Mikrocontroller gespeichert
 - IR-Verbindung zum PC

ARCADEmini - Hardware



ARCADEmini-Assembler

```

GET_KEYS      $KeyState, $KeyPress, $KeyRel;

JMP_AND_NZ   @DezInit, $KeyPress, 0x01;
JMP_AND_NZ   @BcdInit, $KeyPress, 0x02;
JMP_AND_NZ   @BinInit, $KeyPress, 0x04;
JMP_AND_NZ   @SetTimeHour, $KeyPress, 0x20;
JMP_AND_NZ   @SetTimeMin, $KeyPress, 0x40;
    
```

Blinkenlights-Markup
-Language



Blinkenlights-Movie



blm2ambin →

ARCADEmini-Binary



← bmm2ambin

BLINKENmini-Movie



BLP – Blinkenlights-Protocol

- jedes Frame ein UDP-Paket
- schwarz/weiss, 1 Kanal

Dynamische Erweiterung

- periodische Anfrage
- z.B. alle 10s

magic			
0xDE	0xAD	0xBE	0xEF
frame number			
0x00	0x00	0x00	0x00
width		height	
0x00	0x12	0x00	0x08
pixels			
0x01	0x01	0x00	0x00
0x00	0x00	0x00	0x00
...			
0x00	0x00	0x00	0x00



- Pixel zeilenweise
- von links nach rechts
- von oben nach unten

magic			
0xDE	0xAA	0xBE	0xCD
command			
'R'	'E'	'F'	'R'
'E'	'S'	'H'	

- Ende-Mitteilung (optional)

magic			
0xDE	0xAA	0xBE	0xCD
command			
'C'	'L'	'O'	'S'
'E'			

EBLP – extended Blinkenlights-Protocol

- jedes Frame ein UDP-Paket
- Graustufen, 1 Kanal

magic			
0xFE	0xED	0xBE	0xEF
frame number			
0x00	0x00	0x00	0x00
width		height	
0x00	0x12	0x00	0x08
pixels			
0xFF	0xCC	0x99	0x66
0x33	0x00	0x00	0x00
...			
0x00	0x00	0x00	0x00



- Pixel zeilenweise
- von links nach rechts
- von oben nach unten

Dynamische Erweiterung

- periodische Anfrage
- z.B. alle 10s

magic			
0xDE	0xAA	0xBE	0xCD
command			
'R'	'E'	'F'	'R'
'E'	'S'	'H'	'2'
'5'	'6'		

- Ende-Mitteilung (optional)

magic			
0xDE	0xAA	0xBE	0xCD
command			
'C'	'L'	'O'	'S'
'E'	'2'	'5'	'6'

MCUF – Microcontroller Frame Unit

- jedes Frame ein UDP-Paket
- Graustufen, mehrere Kanäle

magic			
0x23	0x54	0x26	0x66
height		width	
0x00	0x14	0x00	0x1A
channels		maxval	
0x00	0x01	0x00	0xFF
pixels			
0xFF	0xCC	0x99	0x66
0x33	0x00	0x00	0x00
...			
0x00	0x00	0x00	0x00



- alle Kanäle pro Pixel
- Pixel zeilenweise
- von links nach rechts
- von oben nach unten

- zulässige Werte für Pixel:
0 .. maxval

Dynamische Erweiterung (BlinkenProxy)

- periodische Anfrage
- z.B. alle 10s

magic			
0x42	0x42	0x42	0x42
reserved			
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

- Ende-Mitteilung (optional)

magic			
0x42	0x42	0x42	0x43
reserved			
0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00

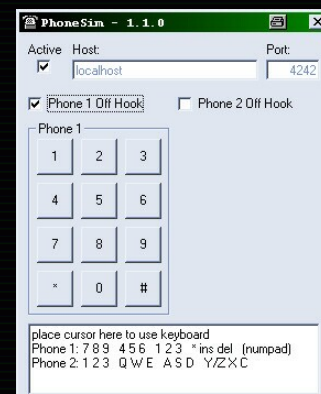
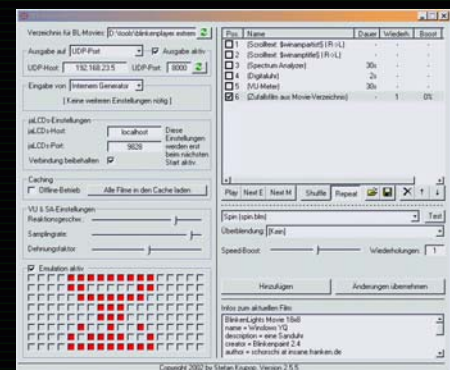
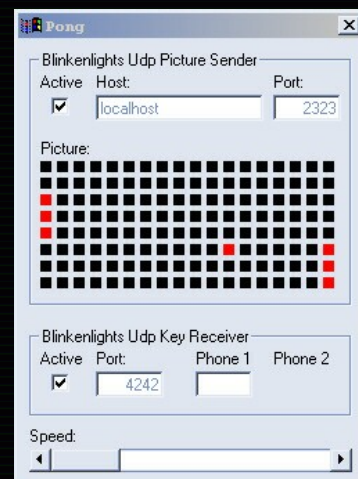
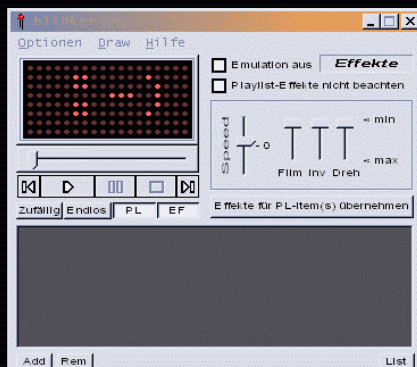
bl_proxy - Blinkenlights-Proxy

- Blinkenlights-Streams ...
- ... empfangen
- ... filtern
- ... konvertieren
- ... auswählen
- ... verteilen



- Unterstützung für BLP-, EBLP- und MCFU-Protokoll
- Unterstützung für dynamische Protokollerweiterungen
- kann auf mehreren UDP-Ports empfangen
- iptables-ähnliche Filterung der ankommenden Streams
- prioritätsbasierte Stream-Auswahl aus den aktiven Streams
- Konvertierung zwischen Formaten (Breite, Höhe, Graustufen)
- Fernsteuerung über TCP-Verbindung möglich

- Blinkenleds Player
- BlinkenTool
- BlinkenPlayer Exxtreme
- Blinkensim VB
- PhoneSim
- Pong, Tetris, Snake
- jaLCDs
- ...



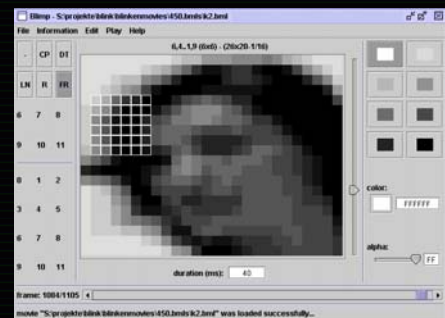
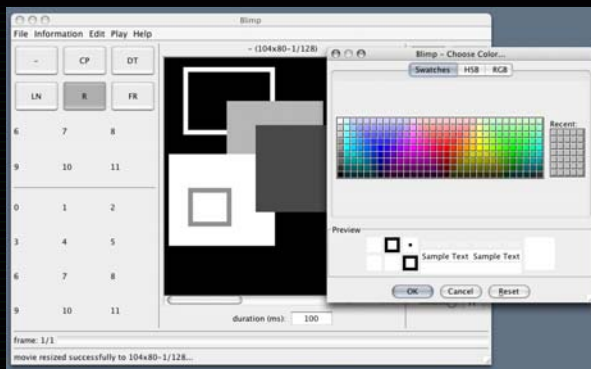
- blib - [Blinkenlights Library] – Spiele
- BlinkenLib
- blccc [Blinkenlights Chaos Control Center]
- blinkensdn
- blinkentools
- blinkensim
- blinkenthemes
- blinkenjava
- udp2telnet
- Kernelmodule für Schieberegister HW
- bmgames
- bmsim
- net2bmdrv

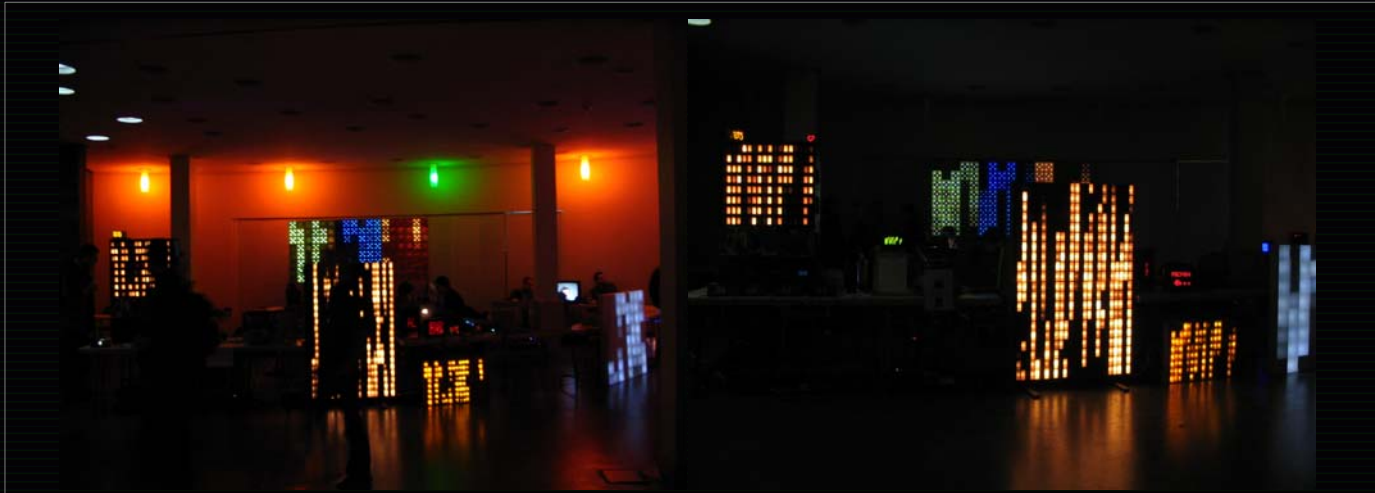


Press ENTER to finish.

```

...
st@blccc: ~/blccc/playlists> blccc localhost
bl_playlist_new: successfully loaded 'playlist.default' with 162 movies
bl_movie_load: successfully loaded '21st_century_man.blm'
isdn_state_changed: onhook onhook
bl_movie_load: successfully loaded '3D_cube.blm'
bl_movie_load: successfully loaded '3rd_advent.blm'
bl_movie_load: successfully loaded '4th_advent.blm'
bl_movie_load: successfully loaded 'alloybase.blm'
bl_movie_load: successfully loaded 'antiwar.blm'
    
```

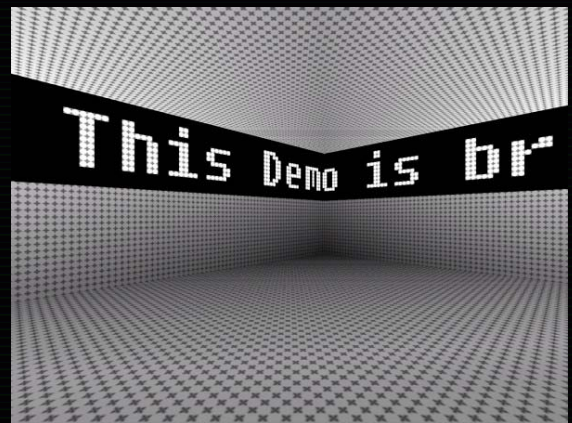




21C3 | Art & Beauty | BlinkenArea



21C3: The Blinking Suspects - Die blinkenden Verdächtigen



shutdown -h now

The Blinking Suspects
2103 | 27. 12. 2004



<https://blinkenarea.org/>

copyleft - all rights reserved