

# Die Pronisation der Dinge

Tobias Rehbein / blabber

21. Februar 2011

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
  
- 2 Das Pronisationsverfahren
  
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
  
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

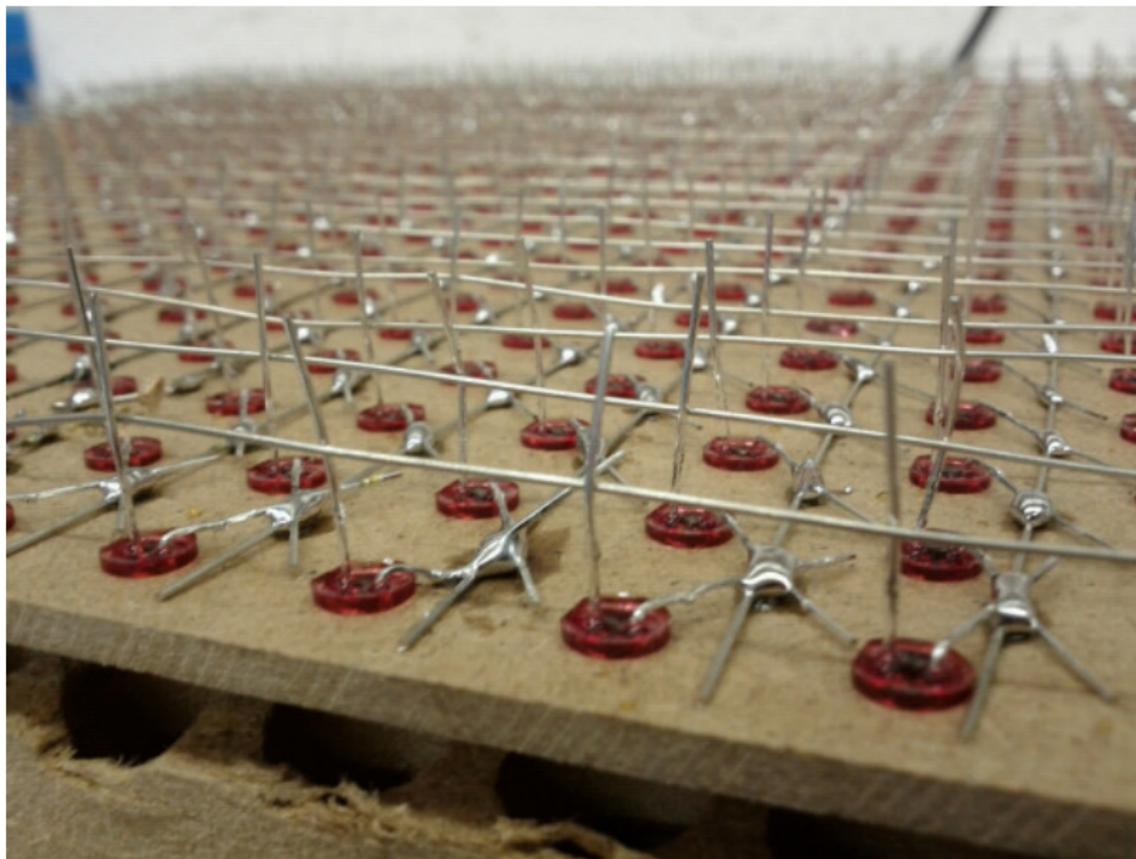
- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

- 1 Motivation
  - PRON-Wall
    - PRON-Wall Protokoll
    - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

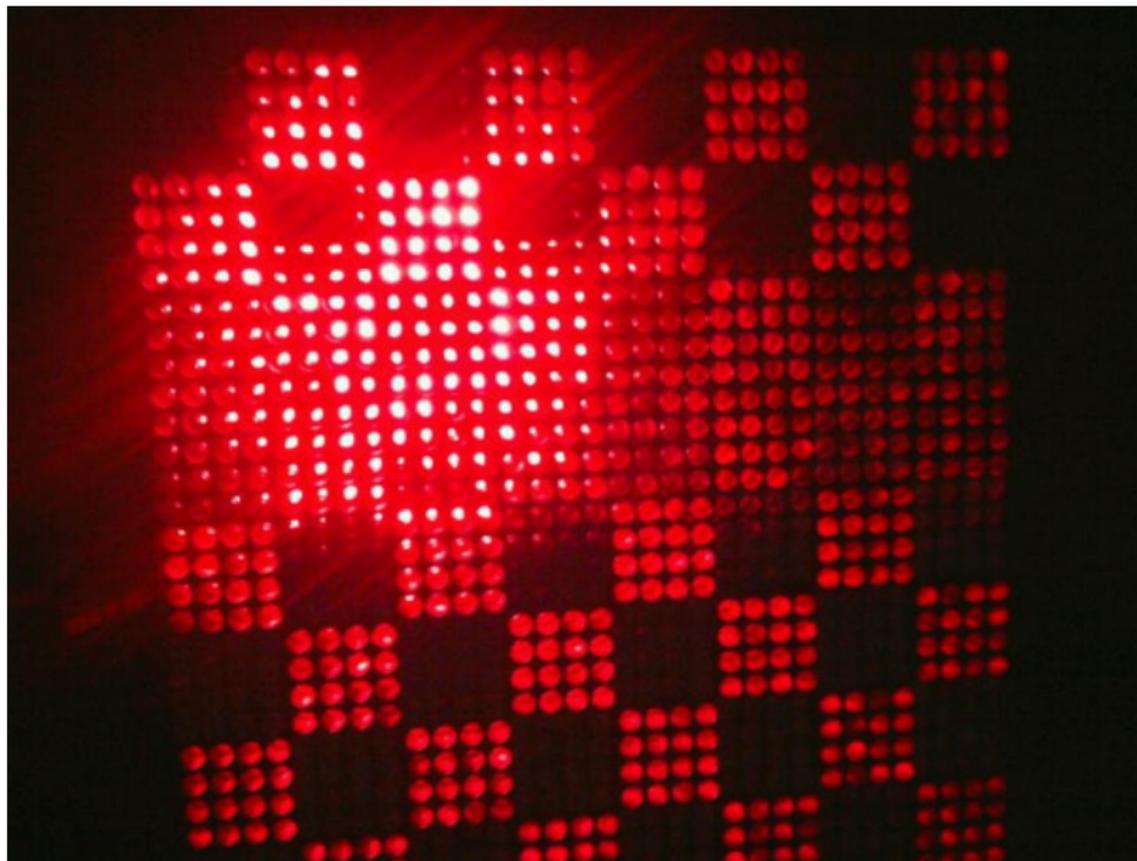
# LED-Panel

- LEDs pro Panel:  $32 \times 32$  (=1024)
- Farbe der LEDs: rot
- Abstand der LEDs (Mitte-Mitte / horizontal & vertikal): 2 cm
- Panelfläche: 64 x 64 cm

# Panel Rückseite



# Panel Testdrive



# LED-Wall

- 8 Panels
- Gesamtfläche: 128 x 256 cm
- Gesamtzahl LEDs: 8192

# Soll- & Ist-Zustand

Gegenwart

Zukunft

# Soll- & Ist-Zustand

## Gegenwart

- ein Panel betriebsbereit
- Laborboard
- zweistufige Animationen

## Zukunft

# Soll- & Ist-Zustand

## Gegenwart

- ein Panel betriebsbereit
- Laborboard
- zweistufige Animationen

## Zukunft

- je Panel ein Controller
- ein zentraler Controller
- Ansteuerung über Ethernet

# Soll- & Ist-Zustand

## Gegenwart

- ein Panel betriebsbereit
- Laborboard
- zweistufige Animationen

## Zukunft

- je Panel ein Controller
- ein zentraler Controller
- **Ansteuerung über Ethernet**

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

# RZL1337 - Pron-Wall Protocol (Juli 2011)

## 4.1 Image Frame

```

0      8      16     24     32
+---+---+---+---+---+
| C | TIME | SEQ |
+---+---+---+---+
|          PIXELS ...
-----...
```

C ... type of frame (see 4.2)

TIME ... duration in milliseconds frame will be displayed

SEQ ... will be used for retransmissions

PIXEL ... each pixel is represented by a single byte, containing the brightness level of this pixel

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - **Pronisation**
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

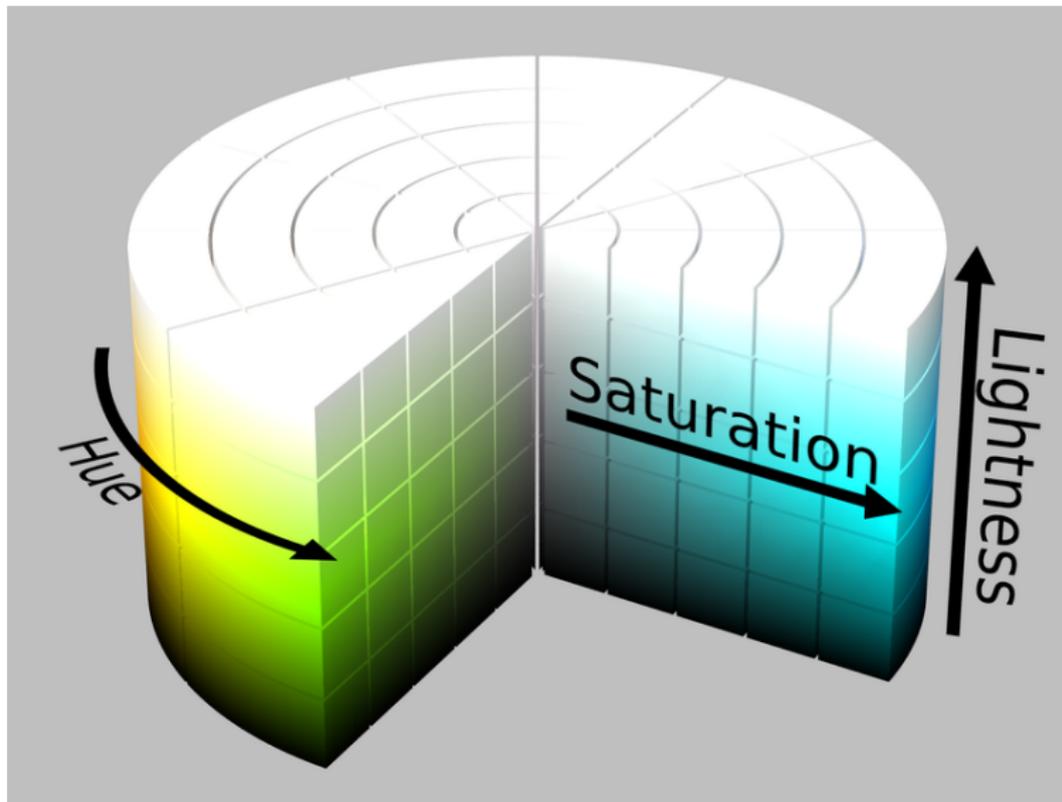
# Definitionen

**Pronisation** bezeichnet den Vorgang der aus einer Grafik einen Bytestrom erzeugt in welchem jedes Pixel durch ein Byte, welches dem Helligkeitswert des Pixels entspricht, repräsentiert wird.

**Pronisat** bezeichnet das Resultat einer Pronisation.

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

## HSL-Farbraum

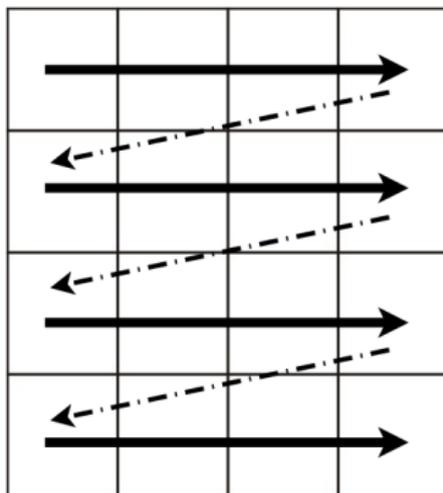


# Verfahren

- zeilenweise, von links nach rechts, jeden Pixel im HSL-Farbraum betrachten
- Lightness als Byte in Bytestrom schreiben (0 = maximal dunkel, 255 = maximal hell)

# Verfahren

- zeilenweise, von links nach rechts, jeden Pixel im HSL-Farbraum betrachten
- Lightness als Byte in Bytestrom schreiben (0 = maximal dunkel, 255 = maximal hell)



# Pronisat



## Pronisat



## Pronisat



- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung**
  - **Fakten**
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

# libpronisate

- Bibliothek implementiert in C
- verwendet die MagickWand API von ImageMagick
- kann Grafiken pronisieren, sonst nichts

# libpronisate

- Bibliothek implementiert in C
- verwendet die MagickWand API von ImageMagick
- kann Grafiken pronisieren, sonst nichts
- erzeugt insbesondere keine Ethernet Frames

# Features

- kann mit Animationen umgehen
- kann über 100 Grafikformate verarbeiten
- kann mit Transparenz umgehen
- skaliert eingelesene Grafiken

# Features

- kann mit Animationen umgehen
- kann über 100 Grafikformate verarbeiten
- kann mit Transparenz umgehen
- skaliert eingelesene Grafiken
- kann über Bande auch mit Blinkenlights Filmen umgehen (blinkentools: b2mng, b2gif)

# Todo

- multi panel support (WIP)
- error messaging
- Farben (?)

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

## Initialisierung

```
pron_init();
```

```
struct pron_context *ctx =  
    pron_context_open("image.png", 32, 32);
```

```
pron_context_close(ctx);
```

```
pron_deinit();
```

## Schleife (Animationen)

```
pron_init();
```

```
struct pron_context *ctx =  
    pron_context_open("image.png", 32, 32);
```

```
for (int f = 0; f < pron_get_frame_count(ctx); f++) {
```

```
}
```

```
pron_context_close(ctx);
```

```
pron_deinit();
```

# Pronisation

```
pron_init();
```

```
struct pron_context *ctx =  
    pron_context_open("image.png", 32, 32);
```

```
for (int f = 0; f < pron_get_frame_count(ctx); f++) {  
    pron_pronisate(ctx, f);
```

```
}
```

```
pron_context_close(ctx);
```

```
pron_deinit();
```

## Pronisat verwenden

```
pron_init();

struct pron_context *ctx =
    pron_context_open("image.png", 32, 32);

for (int f = 0; f < pron_get_frame_count(ctx); f++) {
    pron_pronisate(ctx, f);
    unsigned char *s = pron_get_stream(ctx);
    do_foo(s); /* z.B. per Ethernet versenden */
}

pron_context_close(ctx);

pron_deinit();
```

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - pron-ledwall-designer

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
  
- 2 Das Pronisationsverfahren
  
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
  
- 4 Andere Projekte
  - **libledwall**
  - pron-ledwall-designer

# libledwall

- Lutomas Clientbibliothek für die PRON-Wall
- bisheriger Funktionsumfang: Öffnet einen Raw-Socket auf OSI-Layer 2
- Status: inaktiv

# libledwall

- Lutomas Clientbibliothek für die PRON-Wall
- bisheriger Funktionsumfang: Öffnet einen Raw-Socket auf OSI-Layer 2
- Status: **inaktiv**

# libledwall

- Lutomas Clientbibliothek für die PRON-Wall
- bisheriger Funktionsumfang: Öffnet einen Raw-Socket auf OSI-Layer 2
- Status: **inaktiv**

**Deine Chance einzusteigen**

- 1 Motivation
  - PRON-Wall
  - PRON-Wall Protokoll
  - Pronisation
- 2 Das Pronisationsverfahren
- 3 Die Implementierung
  - Fakten
  - Anwendung (Code)
- 4 Andere Projekte
  - libledwall
  - **pron-ledwall-designer**

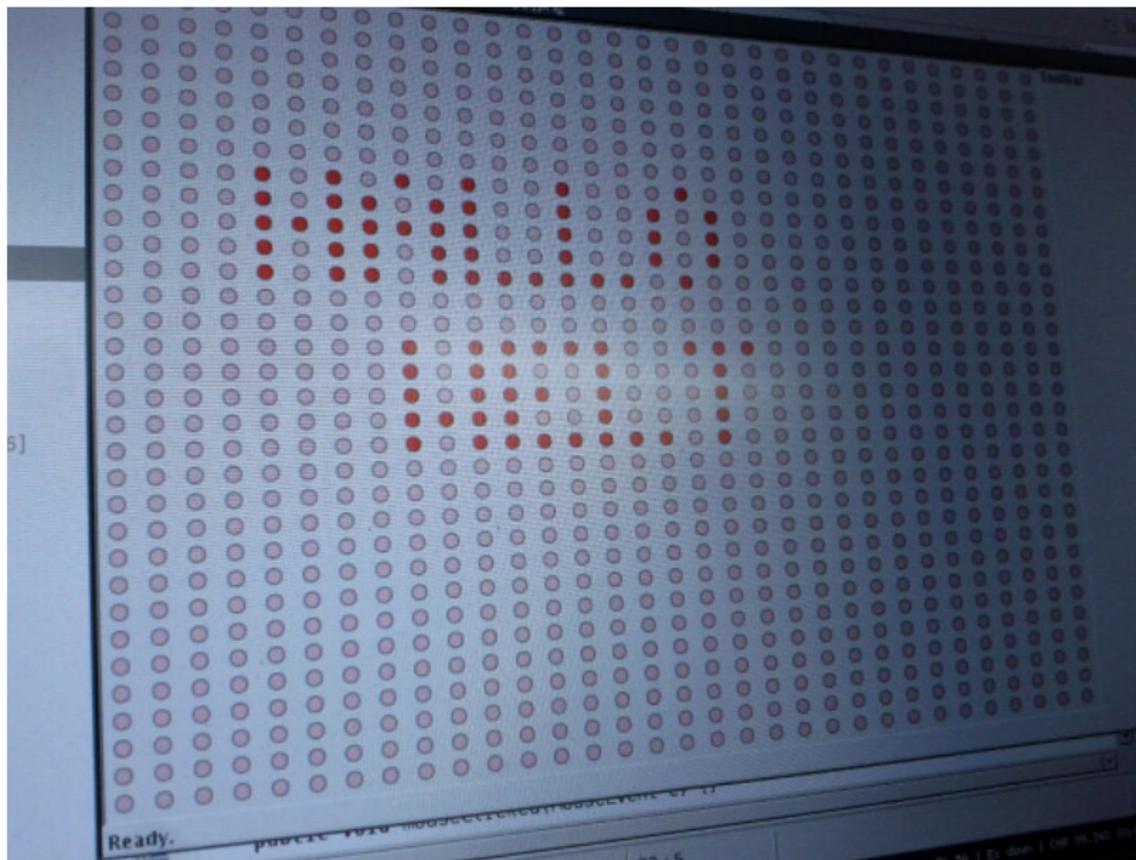
# pron-ledwall-designer

- Elses Designer zur Erstellung und Betrachtung von PRON-Sequenzen
- implementiert in Java
- freie Anordnung der Panels
- Status: ???

# pron-ledwall-designer

- Elses Designer zur Erstellung und Betrachtung von PRON-Sequenzen
- implementiert in Java
- freie Anordnung der Panels
- Status: ???

# Screen



Ende

**Vielen Dank für die Aufmerksamkeit**

[github.com/blabber/libpronisate](https://github.com/blabber/libpronisate)