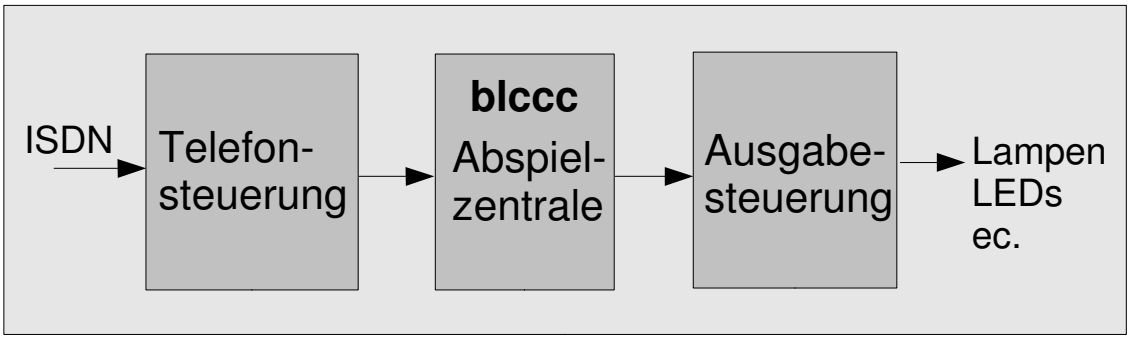
Inhalt **Seite**

1. Einleitung

Das blccc wurde beim Projekt Blinkenlights [1] zum ersten mal eingesetzt. Es steuerte die Playlist mit den Blinkenlights Movies und die Loveletters, weiterhin lausche es auf ankommende Anrufe um dann das Spiel Pong zu starten welches dann auf dem Haus des Lehrers [2] mit bis zu zwei Spielern gespielt werden konnte.

Das blccc wurde von Sven Neumann [3] geschrieben und weiterentwickelt. Die Version 1.99 hat noch weitere Funktionen/Protokolle und wurde bei Arcade [4] eingesetzt. Ich möchte hier auf die Version 1.0 eingehen die bei Blinkenlights verwendet wurde.



Das blccc kommuniziert per UDP und dem Blinkenlights-Protokoll [blp] mit anderen Komponenten. Somit kann blccc mit allen Projekten genutzt werden dessen Hard- bzw. Software dieses Protokoll unterstützen, wie zum Beispiel auch BlinkenMini [5], BlinkenLEDs [6] und LittleLights [7].

[1] http://www.blinkenlights.de/
[2] http://www.bcc-alex.de/
[3] sven@gimp.org
[4] http://www.blinkenlights.de/arcade/
[5] http://www.haecksen.org/~sphaera/blinkenmini/
[6] http://www.blinkenleds.de/
[7] http://www.littlelights.de/

2. Blinkenlights - Protokoll

Das Blinkenlights Protokoll wird benutzt um monochrome Blinkenlights Filme mittels UDP über ein Netzwerk zu versenden. Jedes Frame des Filmes wird in einem eigenen Paket versand. Die Zeit zwischen den Paketen ist auch die Zeit zwischen den Frames.

Hier noch einmal eine kurze Beschreibung wie ein Blinkenlightsfilm aufgebaut ist:

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
000000000000000000	000000000000000000	011111111000000000
000000011100000000	000000100100100000	100010100100000000
000000100010000000	000000010101000000	100001111111111100
000000000110000000	000000001110000000	100001111111101110
000000001100000000	000111111111111000	100010100100011100
000000000110000000	000000001110000000	011111111000000100
000000100010000000	000000010101000000	000000000000011010
000000011100000000	000000100100100000	000000000000101001

Das Format der Filme ist ASCII und man kann sich die Dateien in jedem ASCII Editor ansehen. Hier der Anfang des littlelights.intro.blm:

```
# BlinkenLights Movie 18x8
# name = LittleLights
# description = LittleLights Intro
# creator = Blinkenpaint 2.4
# author = ST
# width = 18
# height = 8
# loop = no
# duration = 27450

@1000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000

@1000
000000000000000000
000000011100000000
000000100010000000
000000000110000000
000000001100000000
000000000110000000
000000100010000000
000000011100000000
...

...
@100
000000000000000000
000000000000000000
000000011100000000
000000100010000000
000000000110000000
000000001100000000
000000000110000000
000000100010000000

@100
000000000000000000
000000000000000000
000000000000000000
000000011100000000
000000100010000000
000000000110000000
000000001100000000
000000000110000000

@100
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000011100000000
...

...
@100
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000011100000000
000000100010000000
000000000110000000

@100
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000011100000000
...
```

Nachdem der Aufbau eines blms bekannt ist, kann man sich ansehen wie ein Packet mit dem Blinkenlightsprotokoll aussieht. Ein Frame ist nach folgenden Regeln aufgebaut:

- *magic*: ist ein DWord [0xDEADBEEF] welches das Paket als Blinkenlights-Protokoll-Paket identifiziert
- *frame_no*: gibt die laufende Nummer des Frames als DWord an
- *width*: Breite des Frames in Pixeln [Word]
- *height*: Höhe des Frames in Pixeln [Word]
- *line ... line*: hier werden 144 Byte angegben, die die Pixel zeilenweise von links nach rechts und von oben nach unten angeben. 0x00 für einen ausgeschaltenen Pixel und 0x01 oder etwas anderes ungleich 0x00 für einen eingeschaltenen Pixel.

Alle Zahlen werden in network-byte-order (big-endian) angegeben.

Word = 2 Byte
DoubleWord = DWord = 4 Byte

ein Beispiel:

---

magic = 0xDE 0xAD 0xBE 0xEF
frame_no = 0x00 0x00 0x00 0x01 // 1. Frame
width = 0x00 0x12 // 18 Spalten
height = 0x00 0x08 // 8 Zeilen

0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

Dieser Frame würde übertragen werden:

001000000000000000
001000000000000000
001000000000000000
111111111111111111
001000000000000000
001000000000000000
001000000000000000
001000000000000000

3. Installation & Einsatz

Voraussetzungen:

funktionierendes Linux
funktionierende Netzwerkkarte
gcc (compiler)
make (Befehl: make)
glib-2.x (benötigt gettext und pkgconfig)
blib (Blinkenlights Library)

Auf meinem Testrechner habe ich folgendes benutzt:

SuSe Linux 8.0 Minimal
Compiler: gcc 2.95.3
make 3.79.1
pkgconfig 0.15.0
gettext 0.12.1
glib 2.2.0
blib 1.0.1

Sind alle Voraussetzungen erfüllt und die Librarys installiert, kann man sich ans compilieren und installieren von blccc machen.

Zu erst das Tar-Archiv auspacken:

tar zxvf blccc-1.0.tar.gz

nun in das ausgepackte Verzeichnis wechseln:

cd blccc-1.0 und **./configure** ausführen.

Hat dies fehlerfrei funktioniert, weiter mit **make** und danach **make install**.

Mit **make clean** können danach nicht mehr benötigte Dateien entfernt werden.

Nun ist blccc auf den System verfügbar.

Wenn blccc gestartet wird sucht es nach einer Datei namens default.playlist. In dieser Dateii müssen die relativen Pfade der blms stehen die abgespielt werden sollen. Im einfachsten Fall legt man sich ein Verzeichnis mit blms an. Mit **ls > default.playlist** erstellt man sich dann in diesem Verzeichnis seine playlist. Nun kann man mit **blccc IP-Adresse** die Filme an den Ausgaberechner schicken. Wenn man keinen Ausgaberechner hat und die Daten mit blinkensim visualisieren möchte oder die Ausgabe von derselben Maschine gemacht wird kann man die Packete auch mit **blccc locahost** an sich selbst schicken. Der default Ausgabeport von blccc ist UDP 2323. Wenn blccc arbeitet sieht das etwa so aus:

st@blccc:~/blccc/playlist> 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 'allyourbase.blm'
bl_movie_load: successfully loaded 'antiwar.blm'

Das Chaos Control Center lauscht auf UDP Port 4242 auf eingehende Pakete die den Status von zwei ISDN Kanälen beschreiben. Diese Pakete werden durch eine andere Software generiert. Empfängt es keine Signale so geht es davon aus, dass keine Anrufer vorhanden sind. Bekommt blccc die Meldung das ein Kanal offhook geht so startet es automatisch das Spiel Pong für einen Spieler gegen den Computer. Der Anrufer kann jetzt mit den Tasten des Telefons sein Paddel steuern [2 nach oben und 8 nach unten] oder er gibt *Nummer# ein. Nummer ist eine Zahlenkombination hinter der sich ein Liebesbrief verbirgt, dieser wird nun gestartet. Spielt der Anrufer Pong und es ruft ein weiterer Anrufer an, so geht der zweite Kanal offhook und nun schaltet blccc in den Modus Pong Spieler gegen Spieler um. Sind beide Kanäle onhook, also beide Anrufer haben aufgelegt, wird die Playlist wieder gestartet.