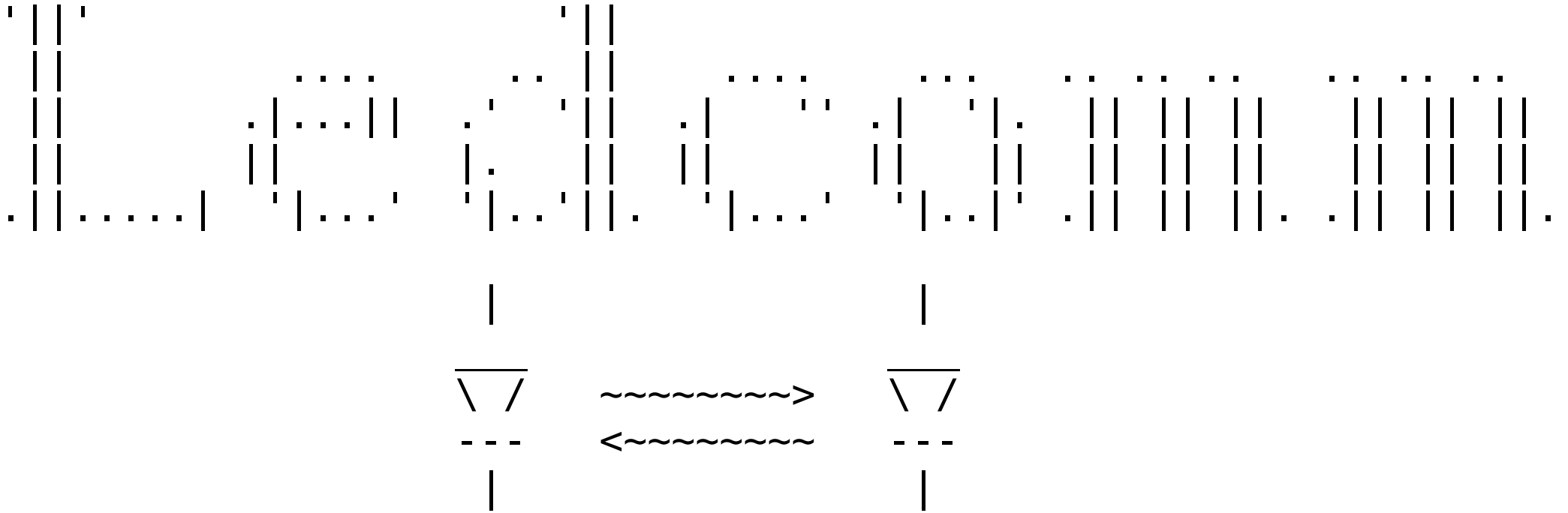


Leuchtdioden als serielle Schnittstellen



(Ledcomm Matthias Koch mecrisp.sf.net [>])

Eine kleine Vorstellung von mir:

Diplom-Physiker mit Nebenfach Gartenbau

Doktorand im Hannoverschen Zentrum für optische Technologien
"Laserspektroskopie an Algen"

Persönliche Interessen:

Obstanbau, Radionavigation, Meeresbiologie,
Historische Tänze, Apnoe-Tauchen, Jonglieren

Programmiersprachen:

Assembler, Forth, Pascal

(Ledcomm Matthias Koch mecrisp.sf.net [->])

Eine Leuchtdiode kann leuchten - aber nicht nur.

Vom Aufbau her ist sie auch eine Photodiode und kann Helligkeit messen.

Die Möglichkeit, schnell hintereinander zu Leuchten und zu Schauen kann mit einem pfiffigen Algorithmus zum Kommunizieren verwendet werden.

(Ledcomm Matthias Koch mecrisp.sf.net [- >])

Geocaching-Reaktivlicht

Hintergrundbeleuchtung ans Umgebungslicht anpassen

Für draußen: Eine innere Uhr nach der Sonne stellen

Fehlermeldungen und neue Brühprogramme über die Power-On-Led der Kakako- und Kaffeemaschine laden

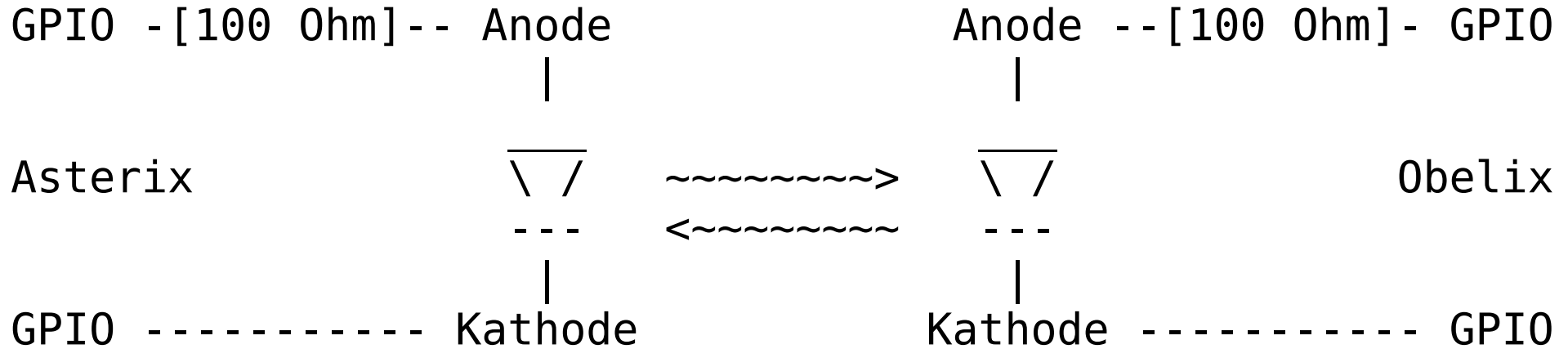
Wasserdichtes Debug-Terminal am Bienenstocktemperaturfühler

Taschenlämpchen als optischer Schlüssel

Glühwürmchen-Blinkmuster-Kunstinstallation

(Ledcomm Matthias Koch mecrisp.sf.net [-->])

Der Aufbau, gleich mit vollständigem Schaltplan:



Für jede Seite also:

- Zwei freie digitale IO-Pins am Mikrocontroller
- Eine passende Leuchtdiode, je nach uC mit Vorwiderstand

(Ledcomm Matthias Koch mecrisp.sf.net [- - >])

Was Ledcomm ist und gut kann:

- * Eine kleine, einfache und sehr günstige serielle Schnittstelle
- * Kommunikation mit Geräten und Basteleien, die ohnehin eine Leuchtdiode und einen Mikrocontroller enthalten

Was Ledcomm nicht ist:

- * Eine schnelle Schnittstelle
- * Ersatz für einen richtigen Photodiodenverstärker am AD-Wandler
- * Konkurrenz zu IrDA
- * Ein Netzwerk oder Bus mit mehreren Knoten

(Ledcomm Matthias Koch mecrisp.sf.net [--->])

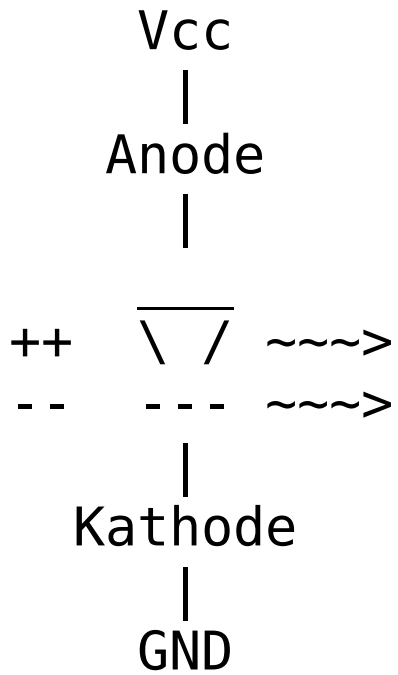
Worüber wir nachdenken müssen:

- * Wie funktioniert die Helligkeitsmessung mit digitalen Pins ?
- * Welche LEDs sind geeignet und warum ?

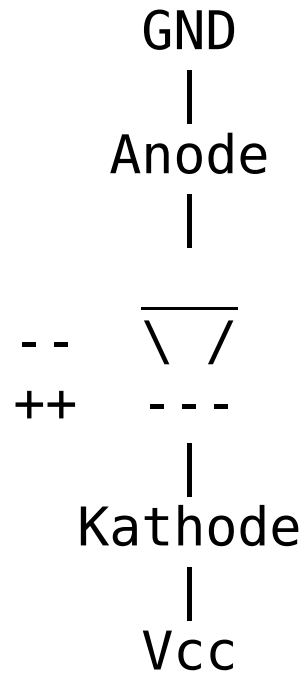
- * Grundlegende Experimente zur Helligkeitsmessung
- * Synchronisation und Verbindungserkennung
- * Lichtpulse senden - und wie sie ankommen
- * Kodierung der Daten
- * Was sieht der Mensch davon ?
- * "Dunkelmodus" und die Möglichkeit,
auch dem Menschen etwas anzuzeigen

- * Fertige Implementationen als Referenz

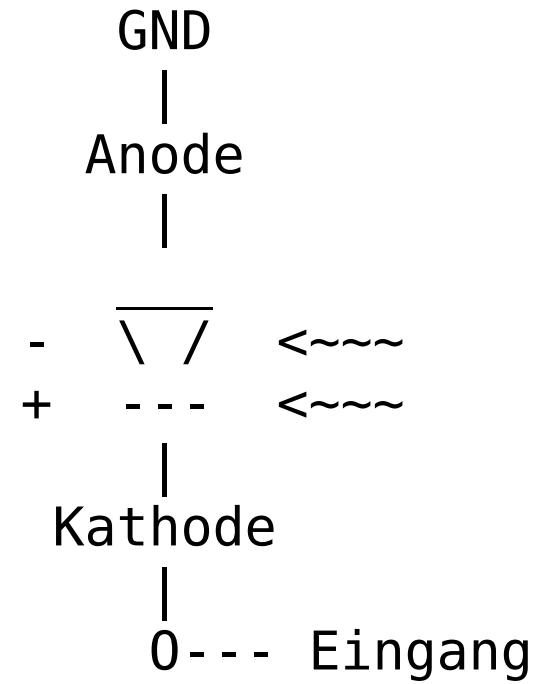
(Ledcomm Matthias Koch mecrisp.sf.net [- - - >])
 Wie funktioniert die Helligkeitsmessung ?



Leuchten



Sperrschichtkapazität
 umgekehrt laden



Photostrom entlädt
 die Leuchtdiode

(Ledcomm Matthias Koch mecrisp.sf.net [---->])

Drei Routinen für die LED:

Leuchten: Anode an Vcc (+), Kathode an GND (-)

Umgekehrt laden: Anode an GND (-), Kathode an Vcc (+)

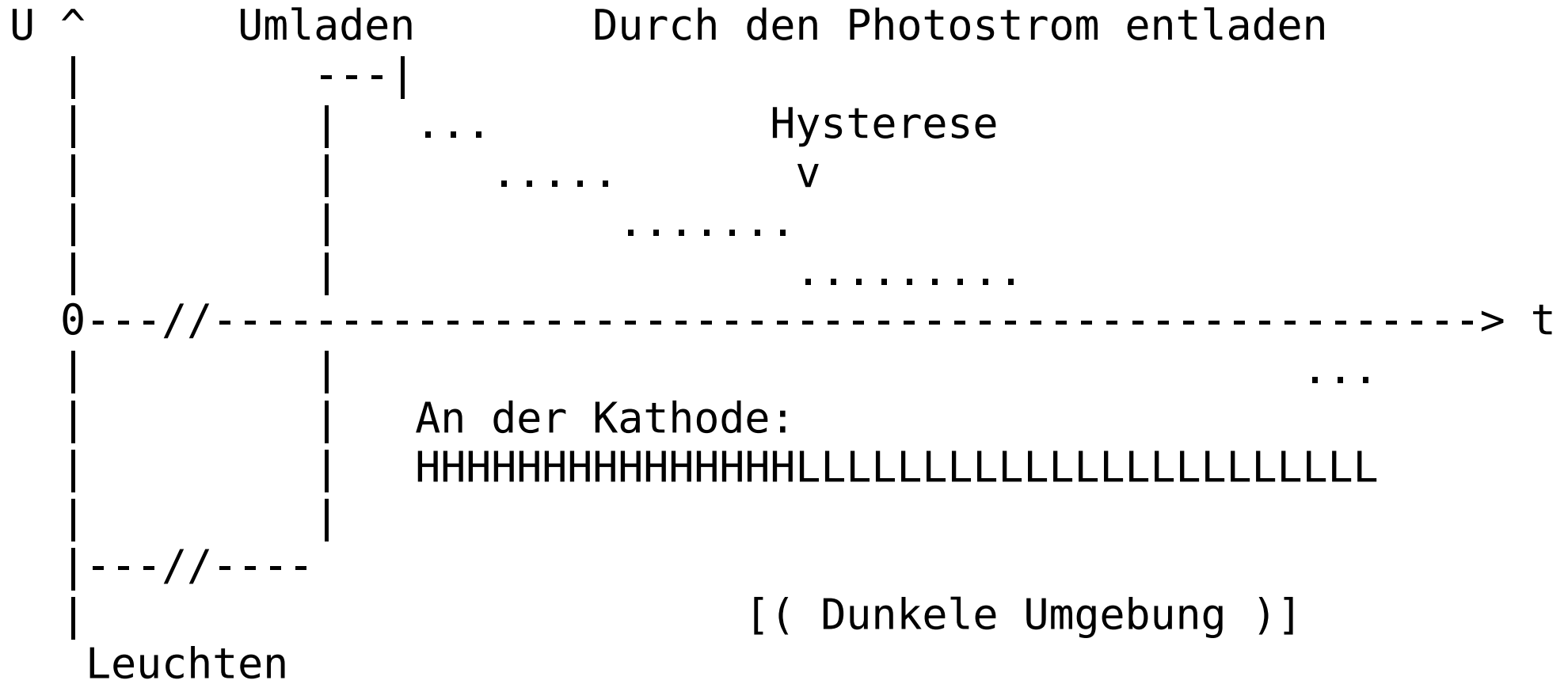
Schauen: Anode an GND (-), Kathode Eingang

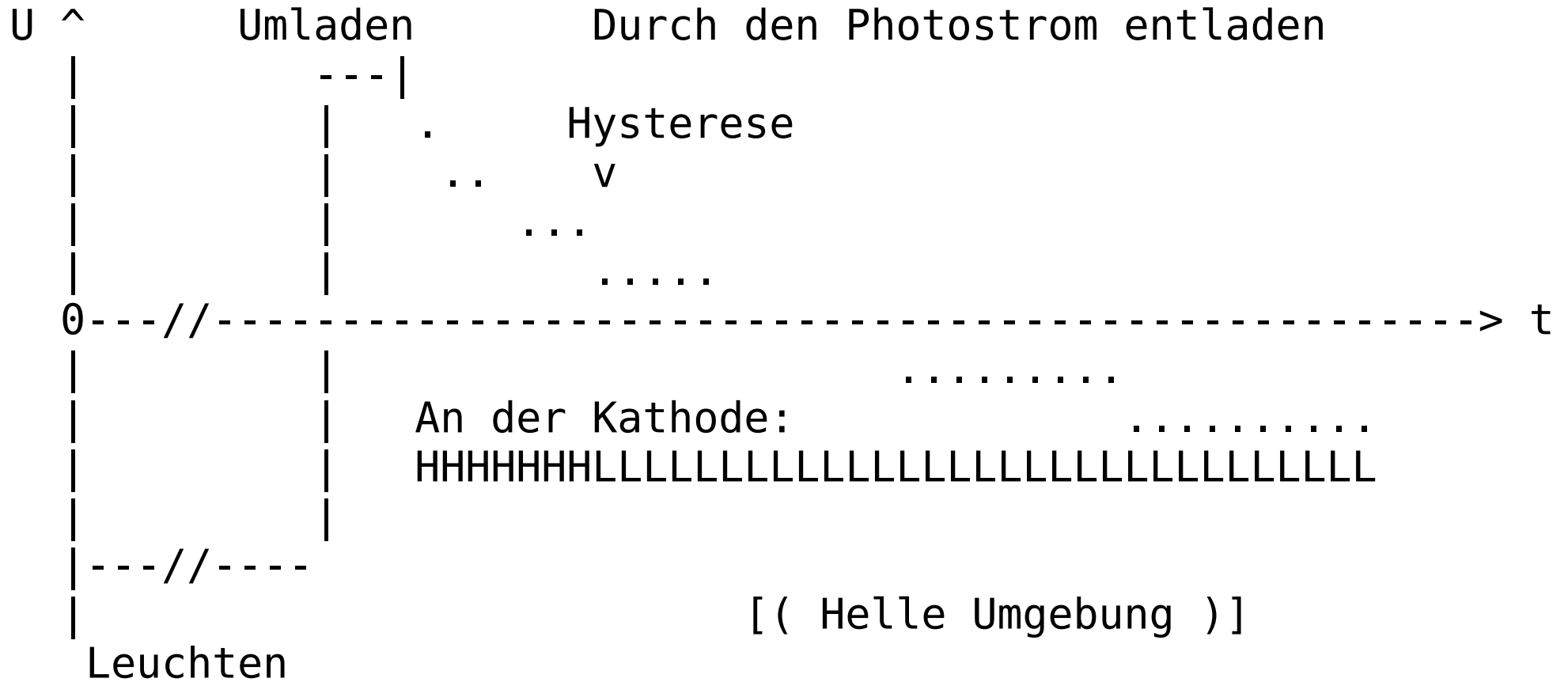
Helligkeitsmessung:

- * Anode Low, Kathode High, ein paar Takte warten
- * Kathode auf Eingang schalten
- * Zeit bis Kathodepin von High auf Low wechselt messen

Je heller es ist, desto größer ist der Photostrom,
umso schneller entlädt sich die Sperrschichtkapazität.

(Ledcomm Matthias Koch mecrisp.sf.net [---->])





(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Welche Leuchtdioden sind geeignet ?

Sie müssen sich gegenseitig sehen können !

Also für ihre eigene Wellenlänge empfindlich sein:

Rote und gelbe LEDs funktionieren gut (Direkte Bandlücke)

Grüne und blaue LEDs nicht (Indirekte Bandlücke)

Hell genug sein und genügend viel Licht empfangen:

Superhelle LED, klares Gehäuse, kleiner Öffnungswinkel

Empfehlung zum Einstieg:

Superhelle, rote, runde 5 mm LED im klaren Gehäuse

(Ledcomm Matthias Koch mecrisp.sf.net [----->])

Was sonst noch wichtig ist:

Die Photoströme sind winzig, die Kapazitäten auch.
Schon kleine Leckströme reichen, dass es nicht funktioniert.

- > Saubere Platinen
- > Kontakte während der Übertragung nicht berühren
- > Kurze Leitungen
- > Oszilloskop ?

Wichtig sind auch hohe Eingangsimpedanzen.

MSP430, ARM Cortex, iCE40 FPGA funktionieren prima
Aber kein NMOS-Z80-PIO :-)

(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Ein kleines Experiment zum Einstieg:

Entladezeiten mit einem Timer messen,
wenn es eine Weile dunkel war und plötzlich hell,
die Leuchtdiode für einen Moment leuchten lassen.

--> Reaktives Licht für Geocaching bei Nacht

Varianten:

Die Messung funktioniert auch, wenn die Anode auf Eingang
geschaltet wird und die Kathode auf Vcc bleibt.
Dann auf den Low --> High Übergang an der Anode warten.

RGB-LED, drei Helligkeitsmessungen zugleich
mit unterschiedlichen spektralen Empfindlichkeiten

(Ledcomm Matthias Koch mecrisp.sf.net [----->])

Vorbereitungen und Vereinfachungen für die Kommunikation:

"Basiszeit" einführen und zwei grundlegende Elemente:

Strahlen: Eine Basiszeit lang leuchten.

Beobachten: Schnell umladen, Basiszeit abwarten, Entladung prüfen
Ergibt ein Bit: Hell oder Dunkel.

Basiszeit so wählen, dass das Umgebungslicht als "dunkel"
und eine Belichtung mit einer gleichartigen Leuchtdiode
als "hell" erkannt wird.

Guter Erfahrungswert, in all meinen Implementationen: 244 us.

(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Allererster Anfang:

8 x Strahlen

32 x Beobachten. Lichtmuster-Bits merken, Flanke suchen:
...-Hell-Hell-Hell-Dunkel-Dunkel gesehen ?
Sofort zurück zum Anfang.

Keine Flanke gefunden ?

Sprung zum Anfang

Damit können sich die Leuchtdioden bereits synchronisieren
und sich abwechselnd "zublinken".

(Ledcomm Matthias Koch mecrisp.sf.net [----->])

Erste Erfolge:

Solange kein Gegenüber erkannt wird, sieht die Schleife dunkler aus (8x Hell, 32x Dunkel) als bei einer bestehenden Verbindung (8x Hell, 2x Dunkel).

Wichtig: "Ungenauer" Takt !

Würden sie exakt synchron laufen,

könnten sich beide vielleicht nie sehen.

Zum Glück ist der interne Oszillator in Mikrocontrollern nicht sonderlich genau, so dass beide auseinanderdriften und sich nach einer Weile sehen können, selbst wenn sie genau gleichzeitig starten.

(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Eine kleine Variation für eine der beiden Seiten:

32 x Beobachten

Flanke gesehen ? 8 x Strahlen, sofort zurück an den Anfang
Zurück an den Anfang

Führt dazu, dass der Knoten im Dunkeln auf eine Verbindung wartet

- Spart Strom
- Synchronisiert immer beim ersten Versuch,
kein "Gleichphasenproblem"
- Hin- und Herschalten zwischen dem hellen und dunklen Modus
erlaubt es, eine Ledcomm-Schnittstelle zugleich
"menschenslesbar" zu gestalten

(Ledcomm Matthias Koch mecrisp.sf.net [----->])

Verbindungsaufbau und Verbindungsabriss erkennen:

Vorbereitung: Synchronisationszähler = 0

8 x Strahlen

32 x Beobachten:

Lichtmuster-Bits merken.

...-Hell-Hell-Hell-Dunkel-Dunkel gesehen ?

Synchronisationszähler < Synchronisationsdauer ?

--> +1. Synchronisationsdauer erreicht ?

--> Verbindungsanfang

Sofort zurück zum Anfang.

Keinen einkommenden Puls erkannt:

Synchronisationszähler = Synchronisationsdauer ?

--> Verbindungsende, Synchronisationszähler = 0

Zurück an den Anfang.

(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Für eine erfolgreiche Synchronisation reichen zwei korrekt erkannte Pulse aus - sie stellen sicher, dass nicht etwas wie das Ausschalten einer Lampe den Verbindungsanfang auslöst.

Es ist immer noch möglich, dass ein flackerndes Licht die Synchronisation auslöst.

Anwendung: LED-Scheinwerfer so modulieren, dass alle beschienenen Ledcomm-Knoten Daten entgegennehmen !
Für Neugierige, tolle Algorithmen zum Stöbern:
--> Fountain-Codes, Vorwärtsfehlerkorrektur

Bei mir: Synchronisationsdauer $18 = 16+2$ beim MSP430
 $34 = 32+2$ beim ARM...

(Ledcomm Matthias Koch mecrisp.sf.net [----->])

Jetzt haben wir bereits:

Synchronisation, Erkennung Verbindungsanfang & Verbindungsende

Lasst uns Daten durch unterschiedlich lange Pulse übermitteln !

Zwei kleine Erweiterungen des Schemas:

- * Nach der Flankenerkennung die Zahl der zusammenhängenden Hell-Bits zählen, um die empfangene Pulslänge zu bestimmen.
- * Eine Routine, die die abzustrahlende Pulslänge bestimmt, die bislang immer 8 gewesen ist

(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Überraschung: Gesendete und empfangene Pulslängen.

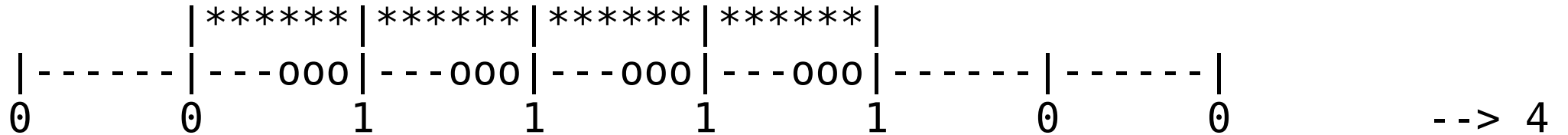
Gesendete Pulslänge: 8 Basiszeiten

Empfange Pulslängen: Je nachdem 7, 8, 9 oder 10 Basiszeiten.

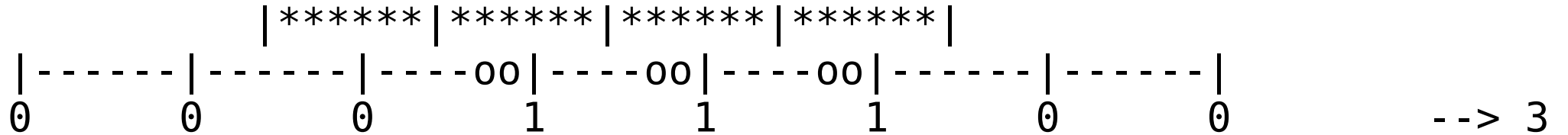
Wie kommt das ?

(Ledcomm Matthias Koch mecrisp.sf.net [----->])

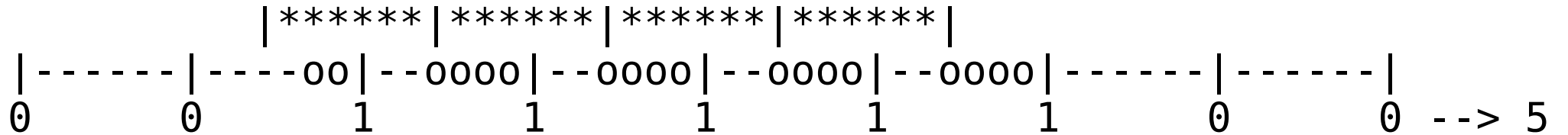
Empfangene Pulslängen bei einem 4-Basiszeiten-Puls:



Eher dunkel:



Sehr hell:



(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Sehr hell + unterschiedlicher Takt:

Kann schonmal als 6 Basiszeiten Dauer ankommen.

Pulslängen können also -1, +-0, +1, +2 ankommen.

Der kürzeste von der Synchronisation erkannte Puls ist zwei Basiszeiten lang.

Damit ergeben sich die sinnvollen Pulslängen:

Abgestrahlt:

4

8

12

...

Kommt an als:

3, 4, 5, 6

7, 8, 9, 10

11, 12, 13, 14

...

Kürzer oder länger: Fehler ?

(Ledcomm Matthias Koch mecrisp.sf.net [----->])

Nun sind wir frei, mit den verschiedenen Pulslängen unsere Daten zu kodieren. Verschiedene Varianten sind möglich !

Meine Lösung verwendet drei verschiedene Pulse:

4 Basiszeiten = "Eins"

8 Basiszeiten = "Null"

12 Basiszeiten = Zeichen fertig

Während der Synchronisation und im Ruhezustand werden Nullen (8) übertragen, ein Datenbyte mit der führenden Eins (4) zuerst und am Ende jedes Zeichens folgt ein längerer Übertrags-Puls (12).

So können kleine Werte mit weniger Pulsen übermittelt werden... Synchronisationsdauer lang genug wählen, dass der Register für ankommende Datenbits "ausgenullt" wird.

(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Und so sieht es aus:

Dunkel:

Aus oder warte im Dunkelmodus auf Synchronisation

Schwaches gleichmäßiges Leuchten:

Warte auf Synchronisation

Helles gleichmäßiges Leuchten:

Verbunden, im Ruhezustand, es werden keine Daten übertragen

Ein kleines bisschen weniger hell, flackert:

Verbindung besteht, Daten werden übertragen

(Ledcomm Matthias Koch mecrisp.sf.net [----->])

Fertige Implementationen, alle kompatibel zueinander:

Assembler: MSP430 (in Mecrisp 1.1.3 enthalten)

Forth: MSP430, ARM Cortex, iCE40 FPGA
(liegt den aktuellen Mecrisp-Paketen bei)

C: ARM Cortex, als Treiber für ChibiOS
<http://klpauba.github.io/LEDComm/>

(Ledcomm Matthias Koch mecrisp.sf.net [----- >])

Leuchtdioden machen Spaß !

```
      ^- )
      ( . . -
        \ \ \
          |>
_____ / | _____ (7 | ` _____ \ | / _____ a:f
```

```
\ Ich freue mich auf viele E-Mails:
\ m-atthias@users.sf.net
\ matthias.koch@hot.uni-hannover.de \
```

12345678901234567890123456789012345678901234567890123456789012345

2

3 297 mm x 167 mm (16:9) Rahmen 10 mm überall.

4 DejaVu Sans Mono Schriftgröße 20

5

6 Format für den Vortrag:

7 17 Zeilen, 65 Buchstaben

8

9

0

1

2

3

4

5

6

7