

QLOCKTWO-Bauanleitung

Diese Anleitung gehört:

Die QLOCKTWO wurde gebaut
vom:

bis:

To-Do's:

Autor: Christian Aschoff, caschoff@mac.com
Version 1.4 vom 1.5.2012



CC BY-NC-SA 3.0:
Namensnennung-
Nicht-kommerziell-
Weitergabe unter gleichen
Bedingungen 3.0

Sie dürfen:



das Werk bzw. den Inhalt
vervielfältigen, verbreiten und
öffentlich zugänglich machen



Abwandlungen und
Bearbeitungen des Werkes bzw.
Inhaltes anfertigen

Zu den folgenden Bedingungen:



Namensnennung - Sie müssen den
Namen des Autors/Rechteinhabers
in der von ihm festgelegten Weise
nennen.



Keine kommerzielle Nutzung - Dieses
Werk bzw. dieser Inhalt darf nicht für
kommerzielle Zwecke verwendet werden.



Weitergabe unter gleichen Bedingungen -
Wenn Sie das lizenzierte Werk bzw. den
lizenzierten Inhalt bearbeiten oder in anderer
Weise erkennbar als Grundlage für eigenes
Schaffen verwenden, dürfen Sie die daraufhin
neu entstandenen Werke bzw. Inhalte nur
unter Verwendung von Lizenzbedingungen
weitergeben, die mit denen dieses
Lizenzvertrages identisch oder vergleichbar
sind.

Vorwort



Diese Bauanleitung beschäftigt sich mit dem Bau einer sogenannten „Wortuhr“. Die Vorlage für diese Wortuhr ist die „QLOCKTWO“ der Firma Biegert & Funk, die als Erfinder dieses Uhrentyps gelten dürften.

Die Firma Biegert & Funk hat mit großer Wahrscheinlichkeit nicht nur einen erheblichen Entwicklungsaufwand betrieben, um diese Uhr zu entwerfen, sondern auch ein Geschmacksmuster auf die Frontplatte oder das ganze Design angemeldet.

Ich möchte daher ausdrücklich darauf hinweisen: Diese Anleitung ist dafür gedacht, eine Wortuhr im QLOCKTWO-Stil für den eigenen Gebrauch „zu Hause“ zu bauen und dabei eine Menge über das sogenannte „physical computing“, also die Verbindung von Kleinstcomputern (Microcontrollern) und der realen, physischen Welt zu lernen. Sie ist nicht dafür gedacht (und daher ist es über die Lizenz ausgeschlossen) irgendwelche kommerziellen Produkte zu unterstützen oder QLOCKTWOs für den Verkauf zu bauen. Ich lehne jedwelche Anfragen im Sinne von „Das Original ist so teuer, können Sie mir nicht eine bauen? Ich bezahle auch das Material!“ ab.

Nach diesen ernsten Worten noch ein paar zu anderen Dingen:

Diese Anleitung geht davon aus, daß der QLOCKTWO-Bauer seinen Kopf einschaltet und eigenständig mitdenkt. Es wird dringend empfohlen, die Anleitung zuerst ganz durchzuarbeiten und erst dann mit dem Bestellen von Bauteilen zu beginnen. Am besten liest man die Anleitung vor dem Bestellen ein zweites Mal durch - so lange, bis sich keine offenen Fragen mehr ergeben. Viele Dinge sind optional oder lassen sich austauschen: die günstigen weißen LEDs durch tolle, hellere „Superflux“ LEDs mit größerem Abstrahlwinkel, die schwarze Folie durch eine rote, den schwarzen Ribba-Rahmen gibt es auch in weiß etc.

Und noch eine Anmerkung zu der Frontplatte: es hat sich herausgestellt, daß der einfachste Weg zu einer schönen Frontplatte eine geplottete Folie aus dem Werbefachhandel ist. Diese Folien sind keine Folien wie man sie in den Laserdrucker oder Tintenstrahldrucker legen kann, sondern Folien, die mit sogenannten Folienplottern geschnitten werden. Das bedeutet, ein Gerät wie eine CNC-Fräse steuert ein Messer, das dann aufgrund der Vektorangaben in der Zeichnung die Folie einschneidet. Aber nur soweit, daß die Trägerfolie nicht mit eingeschnitten wird. Im Prinzip so, als würde man mit einem Skalpell einen Aufkleber oder eine d-c-fix-Folie einritzen. Dann kann man die Folie mit Hilfe einer zweiten Trägerfolie, dem "Application-Tape" auf die Scheibe kleben. Diese Folien können die meisten Digitaldruckereien herstellen. Wenn man mit denen aushandelt, daß man die Buchstaben selbst auslöst, wird es bezahlbar.



Möge die Macht mit Dir sein!

Na dann: Lötkolben vorwärmen und los! Oder wie Yoda sagen würde:

Sicherheitshinweise



Der Bau einer QLOCKTWO ist gefährlich. Teppichmesser, Skalpelle, Glaskanten und Bohrer sind scharf. SMD-Pinzetten und Bohrer sind spitz. Durch das Abwickeln der Beine von Bauteilen fliegen Drahtstücke durch die Gegend. Diese sind schädlich für die Augen.



Während dem Aufbau ist überall ein Haufen Spannung und Strom. Bei einer QLOCKTWO mit Stecker-Netzteil oder Akkubetrieb (die empfohlenen Varianten) ist das kein Problem. Bei dem Einbau eines internen 220 Volt-Netzteils besteht Lebensgefahr.



Lötkolben, Heißklebepistole, Heißkleber und Heißluftpistolen sind genau das: sehr heiß. Reinigungsmittel, insbesondere Isopropanol sind leicht entzündlich.



Lötdämpfe, Reinigungsmittel, Schleifstaub, Kleber, Farbe etc. sind eigentlich immer Gesundheitsschädlich.



Die Matrix wird hochfrequent angesteuert und gibt dabei Störimpulse in Form von Radiowellen ab. Das stört den DCF77-Empfänger, ein Problem, das wir lösen werden. Um andere Dinge, die gestört werden (Herzschriftermacher etc.) muß sich jeder selbst kümmern.



Die Matrix besteht aus sehr hellen „super bright“ LEDs, und davon einer ganzen Menge. Niemals direkt in die LED sehen, insbesondere, wenn während dem Aufspielen der Firmware einzelne LEDs aufgrund von Datenmüll in den Shift-Registern ohne Multiplexing leuchten.

Fazit: Der Bau einer QLOCKTWO ist in jeder Hinsicht gefährlich. Es ist ratsam, eine Schutzbrille, Handschuhe und ein Erste-Hilfe-Set parat zu haben.



Und? Mann oder Memme? Eben. Das Leben als Tüftler und an sich ist gefährlich, also Kopf einschalten, mitdenken, besonnen arbeiten und loslegen!

Versionshistorie

V 1.0	10.3.2011	Erstellt.
V 1.1	23.2.2012	Steuerplatine verbessert, Hinweis auf getrennten Download; Materialliste geändert.
V 1.2	1.3.2012	Appendix C - Öffnungswinkel hinzugefügt.
V 1.3	2.3.2012	Aufspielvorgang auf das RBBB aktualisiert.
V 1.4	1.5.2012	Hinweis auf NachBau_V02-Platine als Appendix D aufgenommen; „Hacking the Firmware“ als Appendix E aufgenommen; kleinere Verbesserungen.

Links

Aktuelle Firmware etc.: <http://code.google.com/p/grosses-bastelwochenende/>

Blog des Autors: <http://christians-bastel-leben.blogspot.com/>

Wyolum (Vertrieb NachBau_V02-Platine und -Kit): <http://wyolum.com>

Arduino: <http://www.arduino.cc/>

Jee-Labs (RBBB): <http://jeelabs.org/>

Reichelt: <http://www.reichelt.de/>

Pollin: <http://www.pollin.de/>

Materialiste



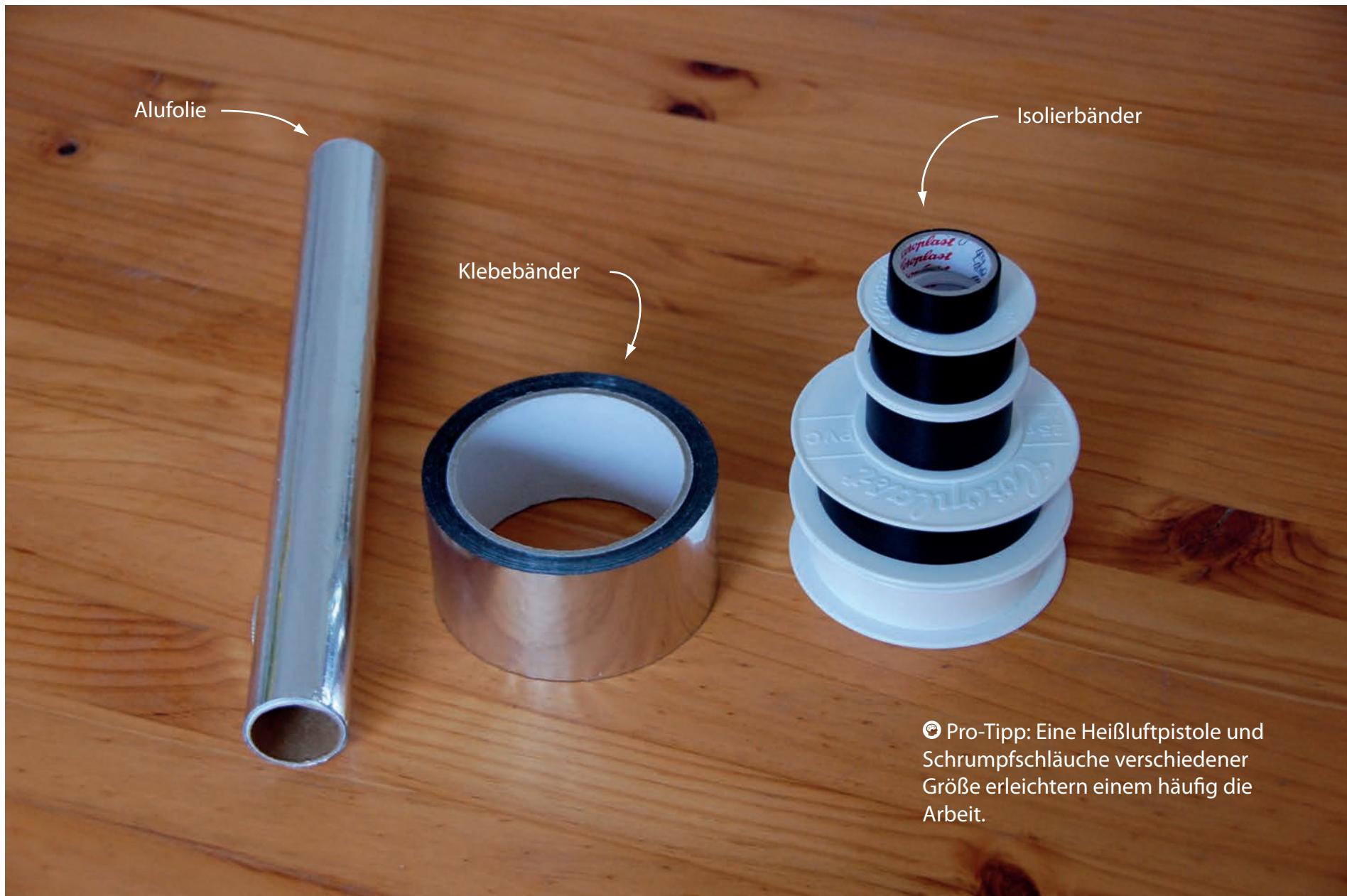
Materialliste (fortgesetzt)



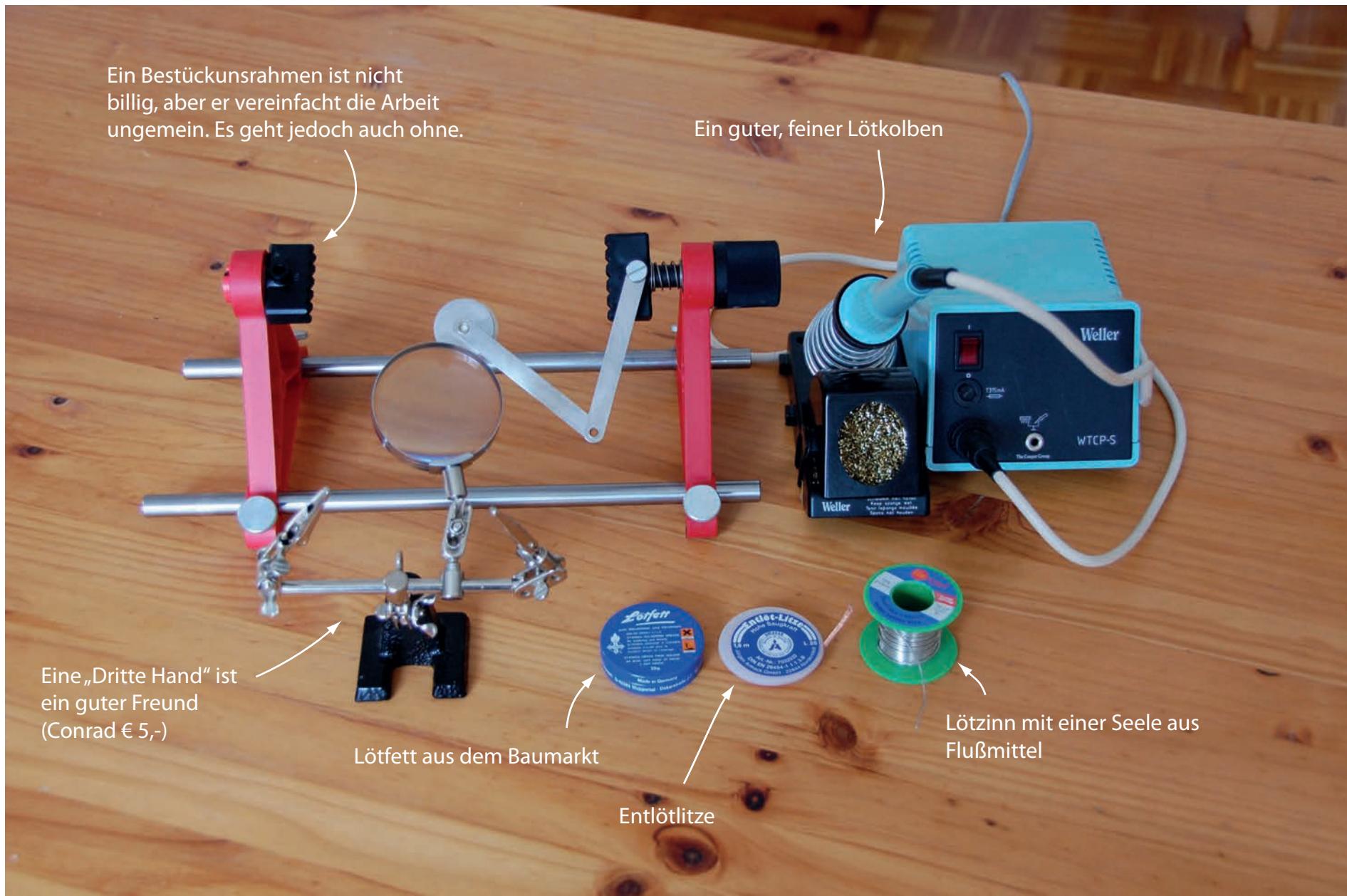
Materialliste (fortgesetzt)



Materialliste (fortgesetzt)



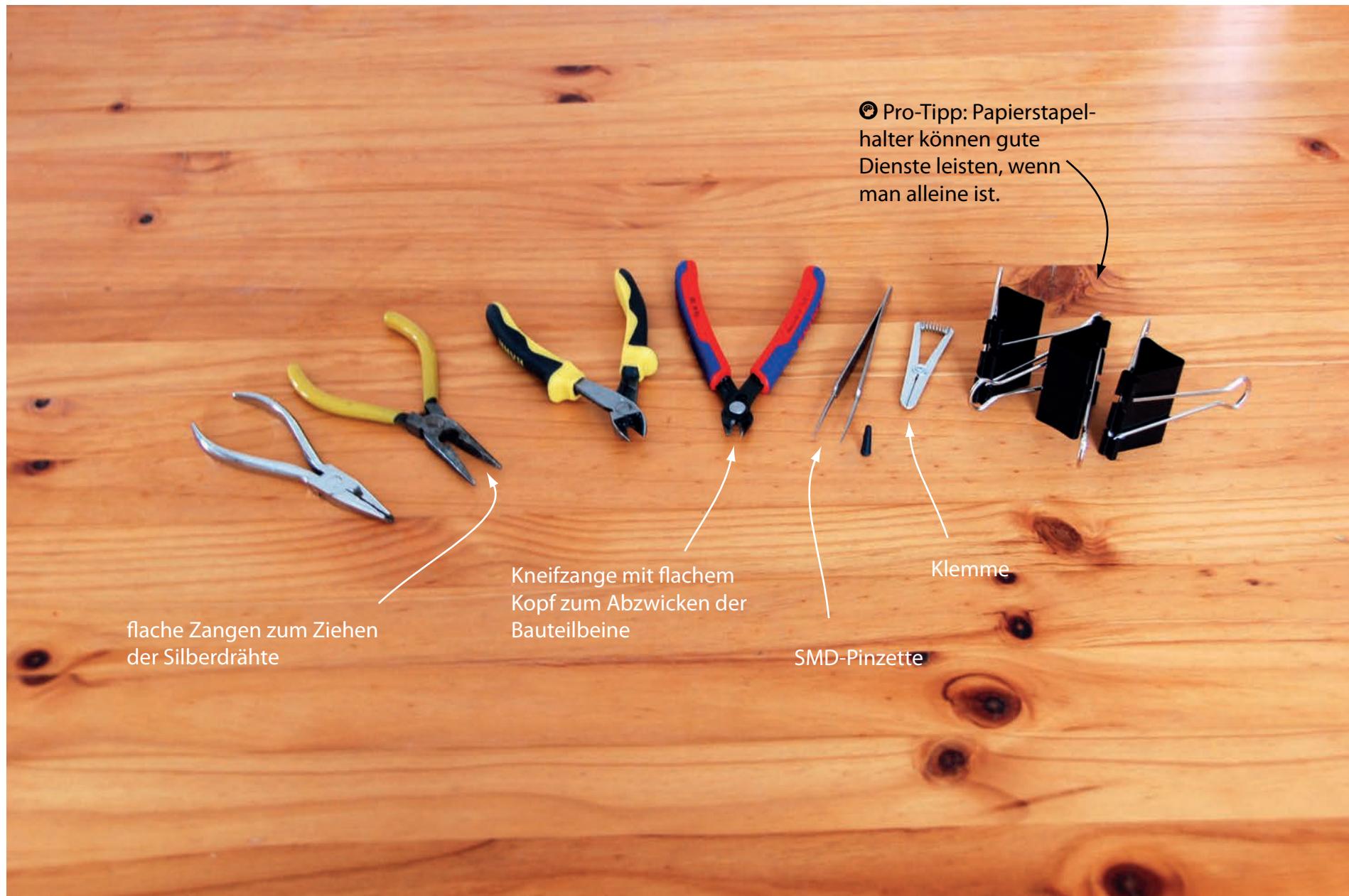
Materialliste (fortgesetzt)



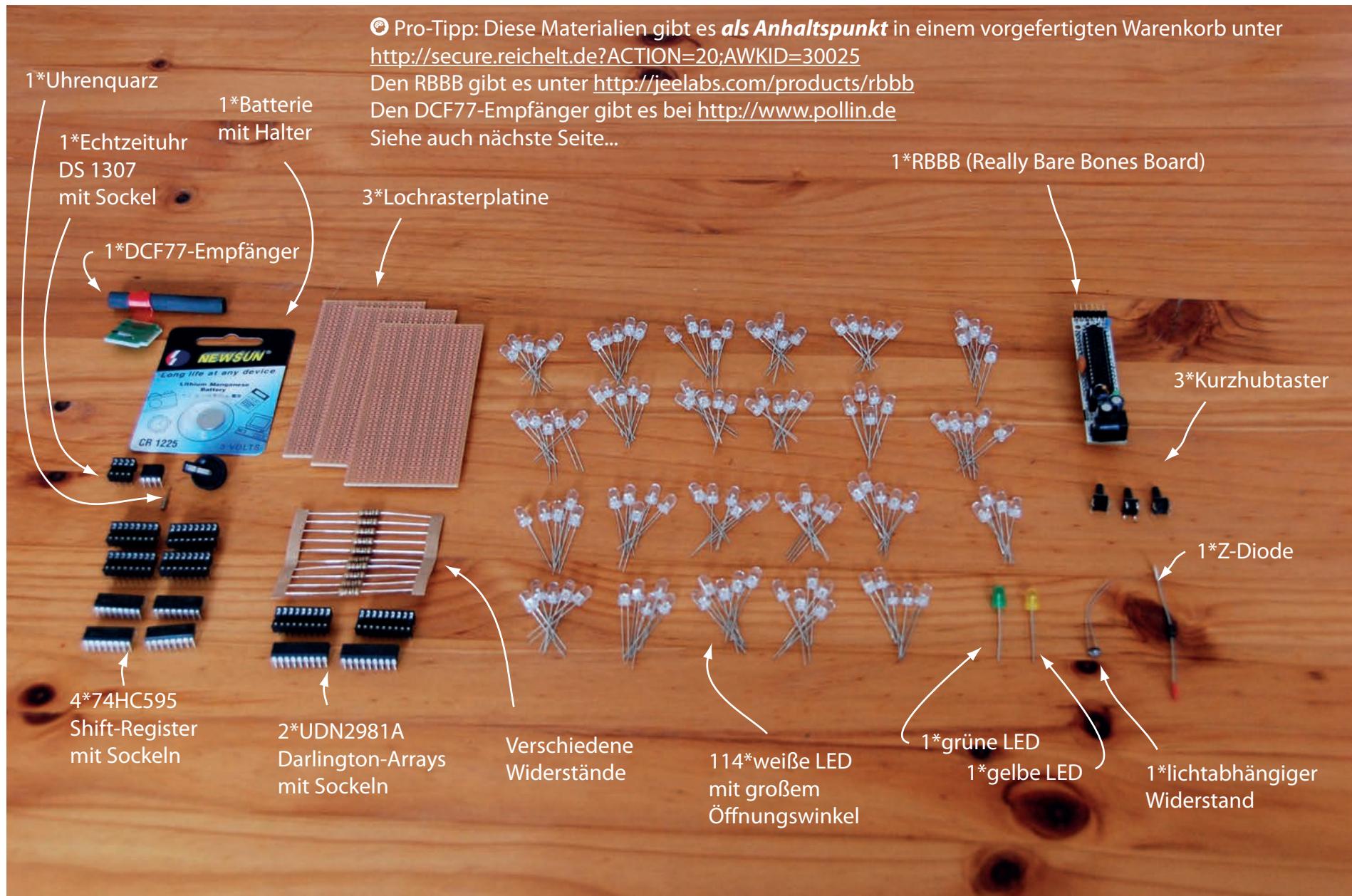
Materialliste (fortgesetzt)



Materialliste (fortgesetzt)



Materialliste (fortgesetzt)



Materialliste (fortgesetzt)

Art. Nr.	Bezeichnung	Einzelpreis	Menge	Preis
0,032768	Mini Uhrenquarz, 0,032768 MHz, TC38	0,15 €	1	0,15 €
1/4W 220	Kohleschichtwiderstand 1/4W, 5%, 220 Ohm	0,10 €	10	0,33 €
74HC 595	IC-SCHALTUNG	0,30 €	4	1,18 €
A 906011	Fotowiderstand	0,95 €	1	0,95 €
APS 300	Stecker-Schaltnetzteil, max 600 mA, "EuP" konform	7,35 €	1	7,35 €
CR 1225	Lithium-Knopfzelle, 3 Volt, 38 mAh, 12,5x2,5mm	0,75 €	1	0,75 €
DS 1307	Real Time Clock I ² C 56B NV SRAM, DIP-8	2,35 €	1	2,35 €
GS 16	IC-Sockel, 16-polig, doppelter Federkontakt	0,050 €	4	0,20 €
GS 18	IC-Sockel, 18-polig, doppelter Federkontakt	0,060 €	2	0,12 €
GS 8	IC-Sockel, 8-polig, doppelter Federkontakt	0,040 €	1	0,040 €
H25PS050	Punkt-Streifenrasterplati. Hartpapier, 50x100mm	0,81 €	3	2,43 €
KZH 12-1	Knopfzellenhalter für Ø 12 mm	1,05 €	1	1,05 €
LED 5-05000 WS	LED, 5mm, ultrahell, farblos, klar, weiß	0,24 €	114	27,36 €
SILBER 0,6MM	Silberdraht, Ø 0,6mm, Länge: 10 Meter	1,75 €	1	3,50 €
SLH 56 GE	LED, 5mm, superhell, eingefärbt, klar, gelb	0,080 €	1	0,080 €
SLH 56 GN	LED, 5mm, superhell, eingefärbt, klar, grün	0,070 €	1	0,070 €
STECKBOARD DBS	Steckbrücken-Drahtbrücken-Set	3,80 €	1	3,80 €
TASTER 3301D	Kurzhubtaster 6x6mm, Höhe: 12,5mm, 12V, vert.	0,16 €	3	0,48 €
UDN 2981 A	Source Driver, DIL-18	1,75 €	2	3,50 €
ZF 3,3	Zener-Diode 0,5W 3,3V	0,040 €	1	0,040 €
1/4W 22K	Kohleschichtwiderstand 1/4W, 5%, 22 K Ohm	0,10 €	10	0,33 €
Gesamt:				56,06 €

Widerstände dem
getrennten Down-
load „Steuerplat-
ne“, „Anodenmulti-
plexer“ und
„Kathodenmulti-
plexer“ entneh-
men.

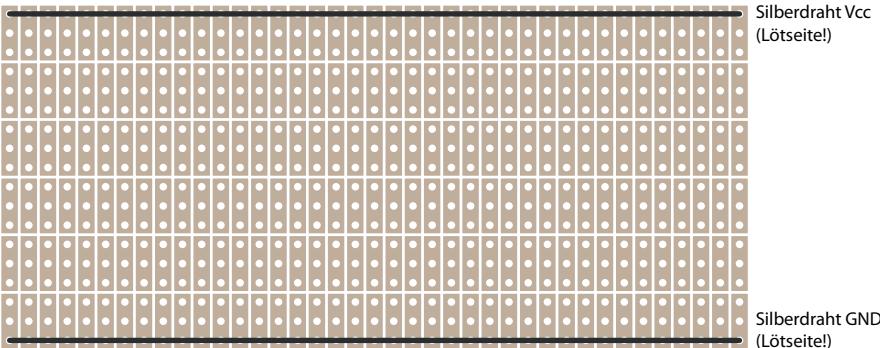
Diese LEDs sind
nicht ideal. Die
LEDs sollten einen
Öffnungswinkel
von min. 70°
haben, siehe
Appendix C!

Die Artikelnummern sind die von der Firma Reichelt. Zusätzlich brauchen wir von Ikea, Pollin und Jee-Labs

810 054	Ribba-Rahmen	15,99 €	1	15,99 €
RBBB	DCF-Empfangsmodul DCF1	4,95 €	1	4,95 €
	Really Bare Bones Board	13,50 €	1	13,50 €
Gesamt:				90,50 €

Dazu kommt dann natürlich noch Kleinkruscht: Fotokarton, Verbrauchsmaterial wie Lötzinn, Litze etc.

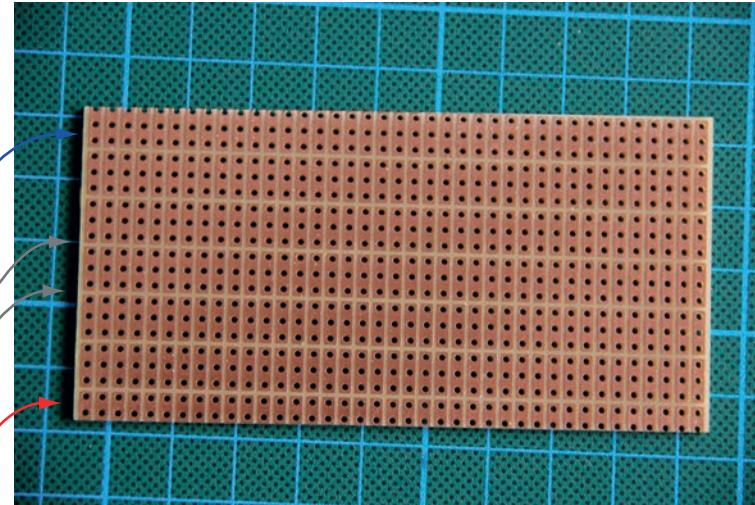
Vorbereiten der Lochrasterplatinen



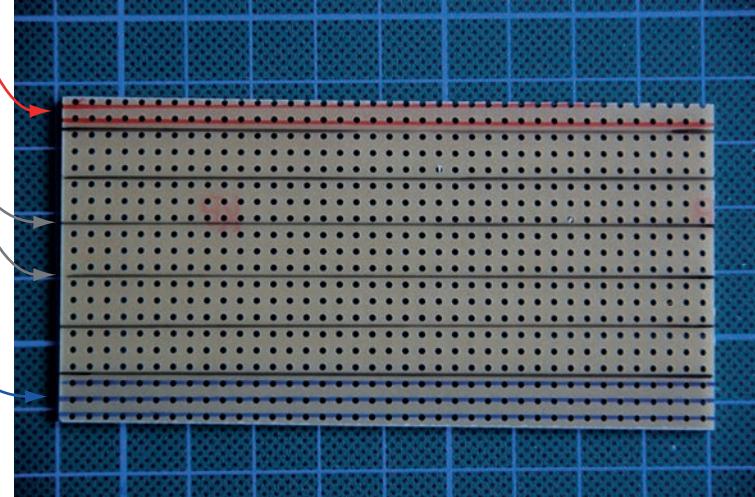
In den schematischen Darstellungen für die Verdrahtung der Platinen sind die Lochrasterplatinen „ideal“ dargestellt. Jede Reihe hat immer drei Löcher.



Damit wir uns nachher einfacher auf der Platinenseite zurechtfinden, die kein Kupfer hat, zeichnen wir uns Markierungen auf die Platine. Dazu eignen sich Permanent-Marker und ein Lineal.

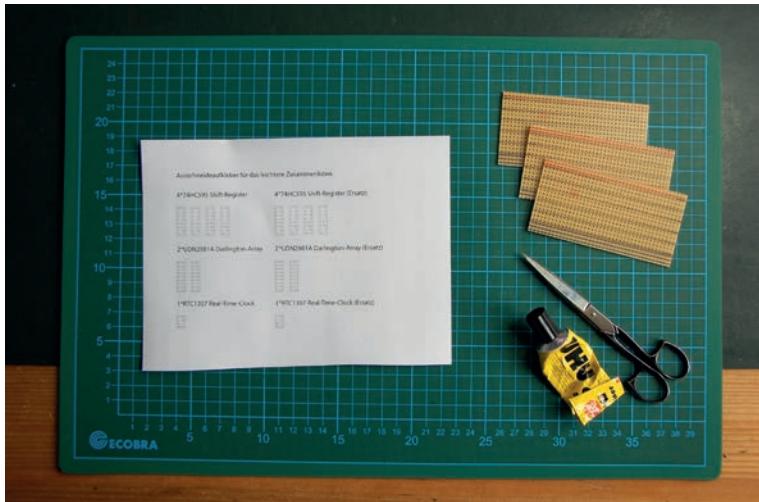


In der Realität sieht das anders aus. Bei dem Beispiel oben hat die obere Reihe fast immer drei Löcher, die untere hat aber nur zwei Lochreihen. Das ist leider bei fast jeder Platine ein bisschen anders.

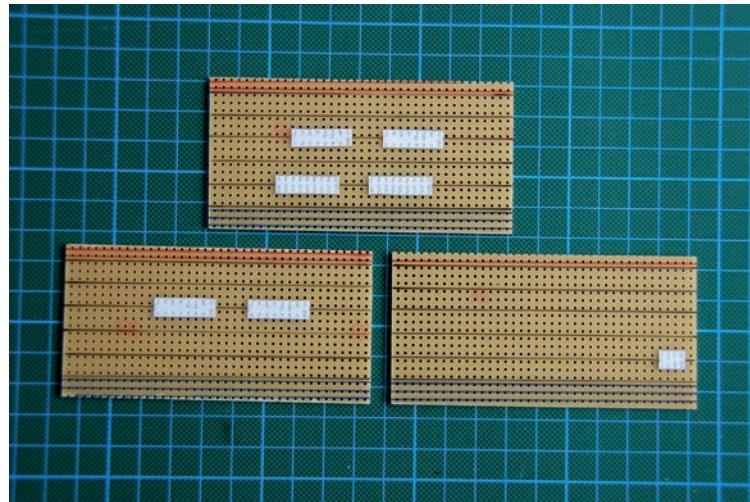


Wir ziehen rote Linien für die Kontaktflächen, die später durch den Silberdraht zu Vcc („Voltage of the common collector“ - Plus) verbunden werden und blaue Linien durch diejenigen, die später mit dem Silberdraht zu GND werden („Ground“, wie der Fachmann sagt). Die dazwischenliegenden Dreierreihen werden mit schwarzen Linien abgetrennt.

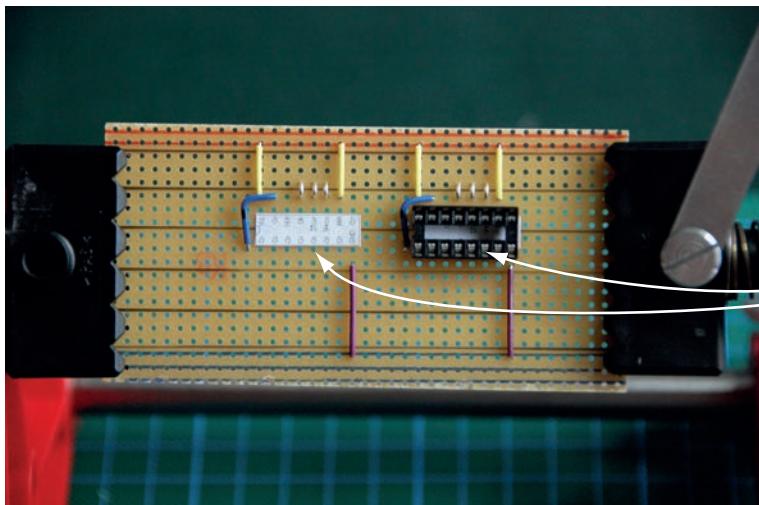
Vorbereiten der Lochrasterplatinen (fortgesetzt)



Damit man sich bei der Verkabelung der Lochrasterplatten nicht vertut, klebt man kleine Chip-Vorlagen auf.



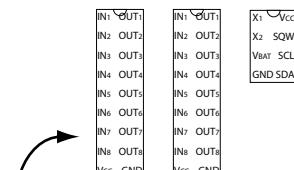
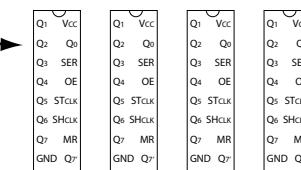
Dafür braucht man die Vorlage vom 74HC595-Shift-Register 4 mal, die von dem UDN-2981A-Darlington-Array 2 mal und die von der DS1307-Echtzeituhr 1 mal. Die Positionen findet man auf den schematischen Verkabelungsplänen von dem Kathoden-Multiplexer, Anoden-Mutliplexer und der Steuerplatine (auf den folgenden Seiten).



Achtung! Die Aufkleber sind kleiner als der Chip, da ja die Beine vom Chip bzw. der Fassung noch außenherum passen müssen.

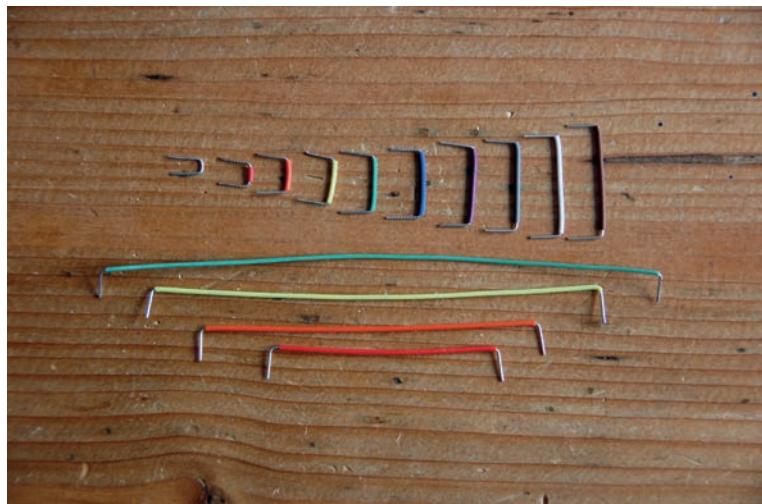
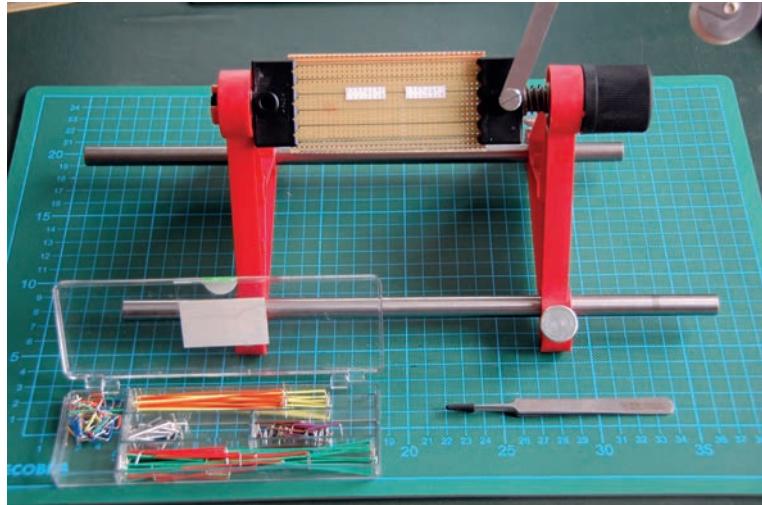
Achtung! Die Aufkleber haben die gleichen kleinen „Nasen“ wie die Chips. Auf die richtige Richtung achten!

4*74HC595-Shift-Register



2*UDN2981A Darlington-Array

Jetzt geht es los!



Ok. Wir sind jetzt so weit, daß wir die erste Platine zusammenlöten können. Um sich irrsinnige Mengen an Abisolierungen und Verzinnungen von Kabelmeterware zu ersparen, kann man für die Verbindung auf den Leiterplatten sehr gut Drähte aus einem Steckbrückensortiment für Steckbretter nehmen.

Außerdem leistet eine SMD-Pinze gute Dienste, um die kleinen Brücken in die Platine zu fummeln.

Wer keinen Bestückungsrahmen hat, sollte eine zweite Person (Freund/-in, Partner/-in, Geschwister, Eltern, Kinder etc.) zur Hand haben.

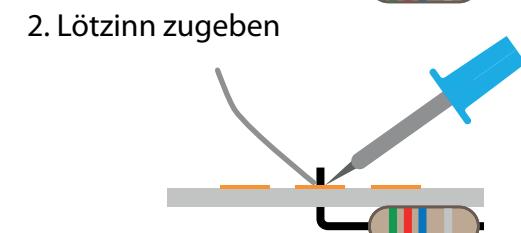
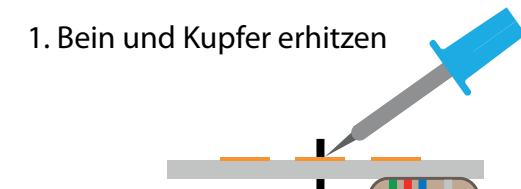
Die Steckbrücken haben verschiedene Längen. Jede Länge hat eine eigene Farbe.

Beim Löten unbedingt darauf achten, daß die Lötstellen gut und sauber sind. Kalte Lötstellen sind im Nachhinein schwer zu finden.

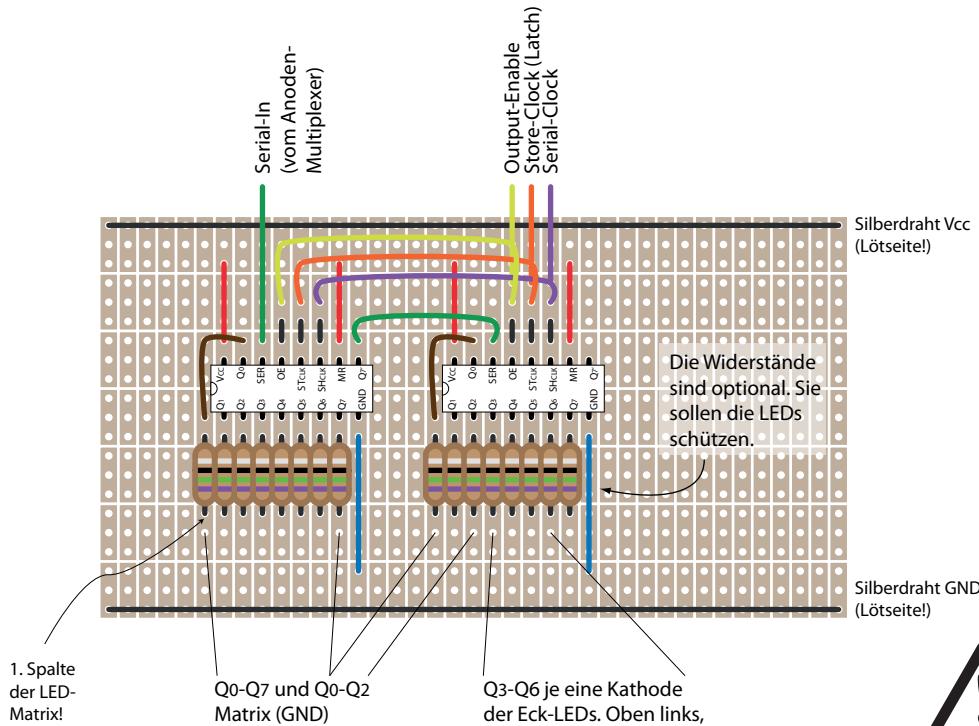
Zuerst Bauteilbein und Platine erhitzen, dann Lötzinn zugeben.

Besser vor dem Zusammenbau doppelt prüfen und gegebenenfalls nachlöten. Die Leiterplatten und Fassungen halten einiges aus. (Die Chips nicht, aber dafür haben wir ja die Fassungen.)

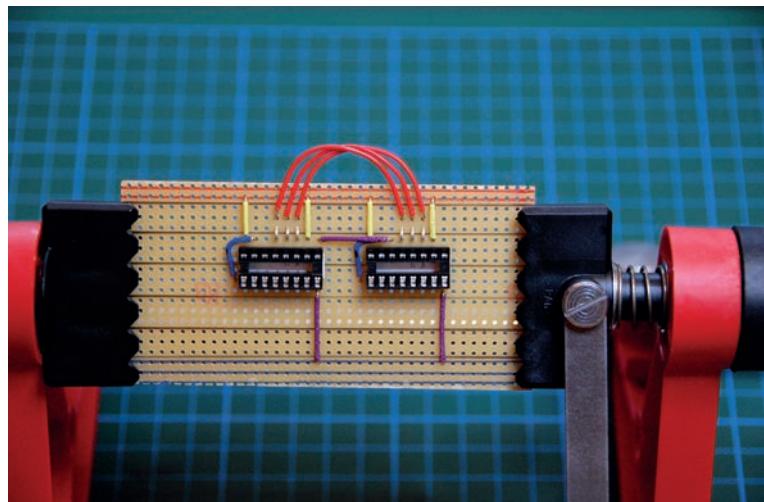
Wenn die Platine fertig ist, die Lötseiten mit Isopropanol und einer Zahnbürste reinigen. Das Flußmittel ist kein idealer Isolator und führt mit der Zeit zu Korrasion. Dieser wichtige Punkt wird von Anfängern gern übersehen.



Der Kathoden-Multiplexer

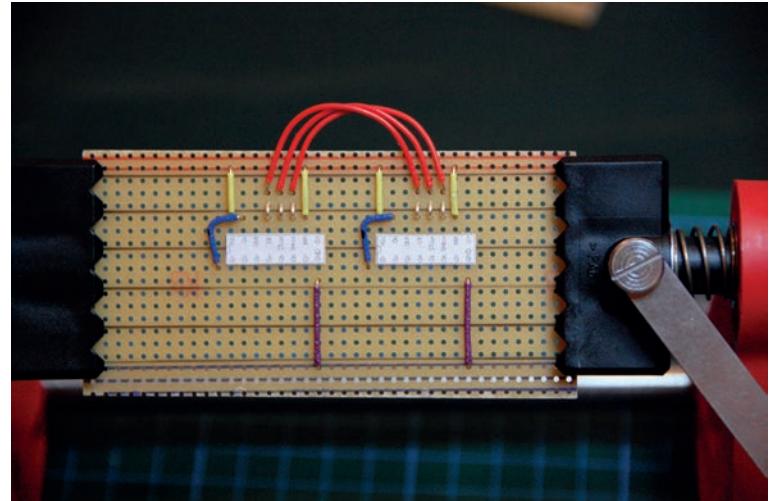


Achtung! Der Kathoden-Multiplexer wird immer mal wieder verbessert. Auf den getrennten Download „Kathodenmultiplexer“ achten.



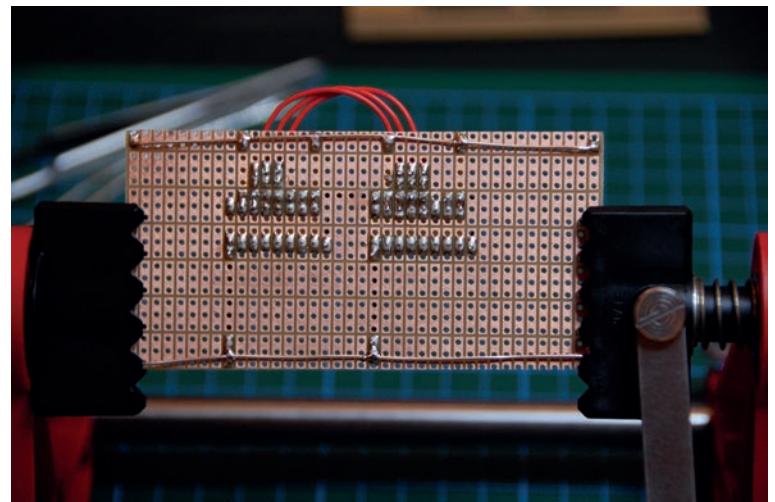
Links: Die Fassungen für die Chips auflöten (Achtung mit der Richtung der „Nase“).

Rechts: Die Rückseite mit den Silberdrähten für Vcc und GND (Ground).

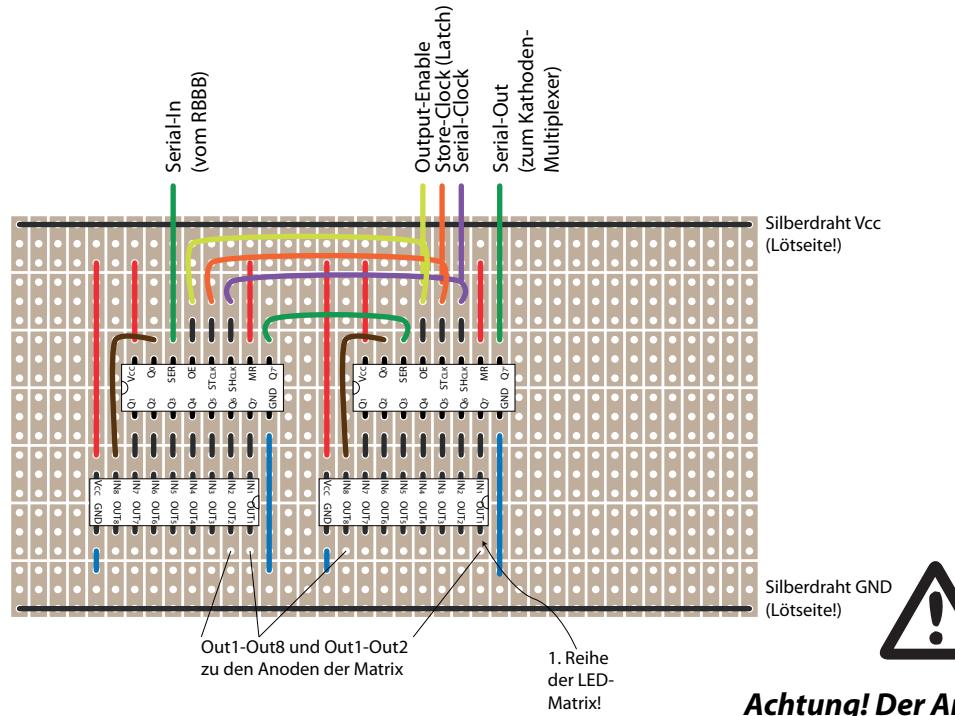


Oben: Zuerst verlötet man nur die Kabel (immer mit den flachsten Bauteilen anfangen).

Die kleinen Verbinder kann man gut selbst aus Silberdraht oder abgezwickten Beinen von Bauteilen biegen.



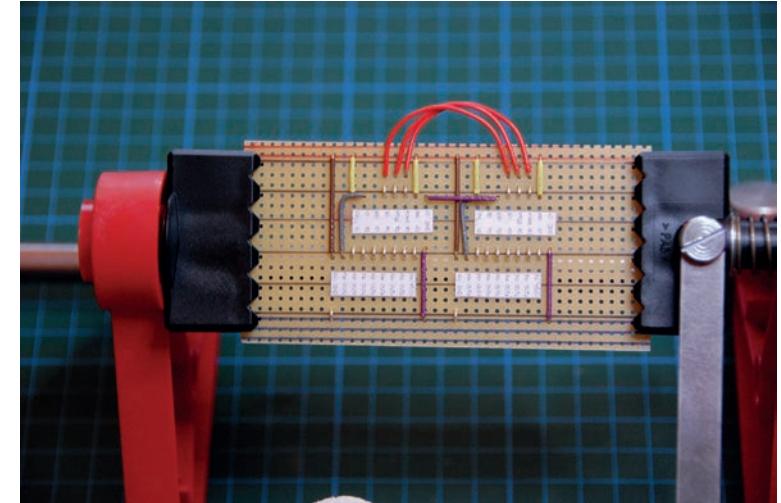
Der Anoden-Multiplexer



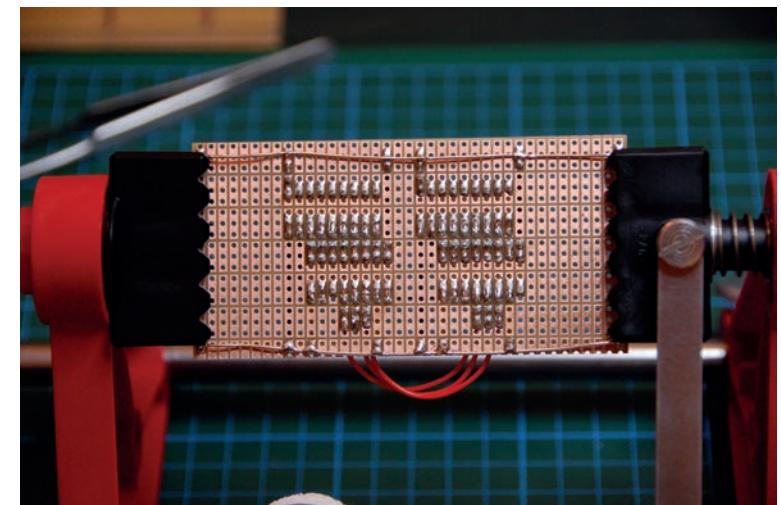
Achtung! Der Anoden-Multiplexer wird immer mal wieder verbessert. Auf den getrennten Download „Anodenmultiplexer“ achten.

Links: Die Fassungen für die Chips auflöten (Achtung mit der Richtung der „Nase“, die Fassungen der UDN2981A-Chips sind um 180° zu den Shift-Registern gedreht).

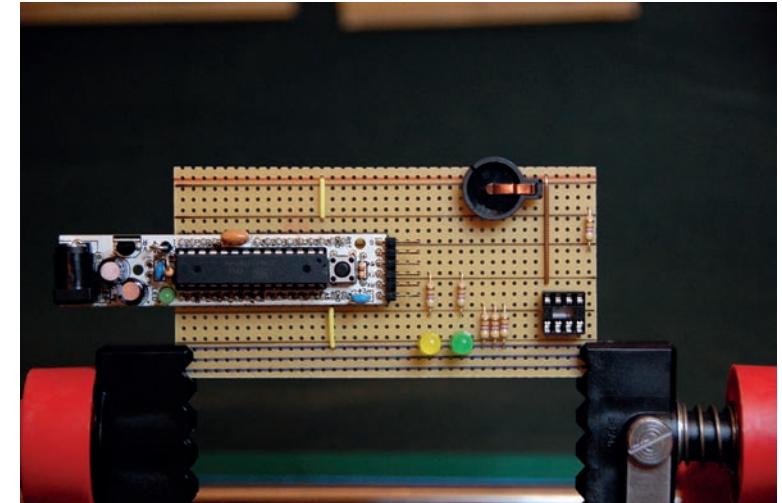
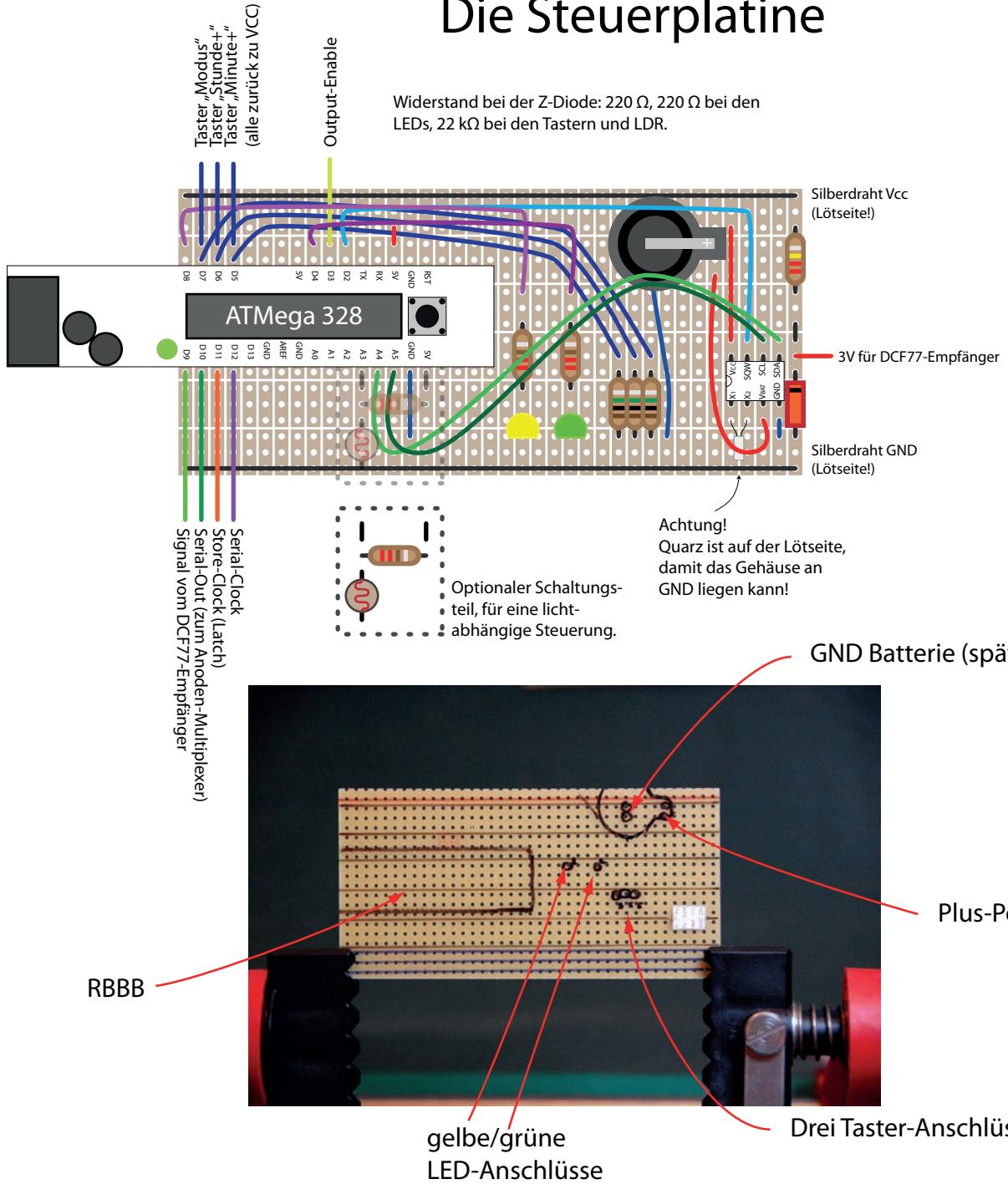
Rechts: Die Rückseite mit den Silberdrähten für Vcc und GND (Ground).



Oben: Der Anoden-Multiplexer ist im oberen Bereich identisch mit dem Kathoden-Multiplexer. Unten kommen dann zur Verstärkung die UDN2981A-Chips dazu.



Die Steuerplatine

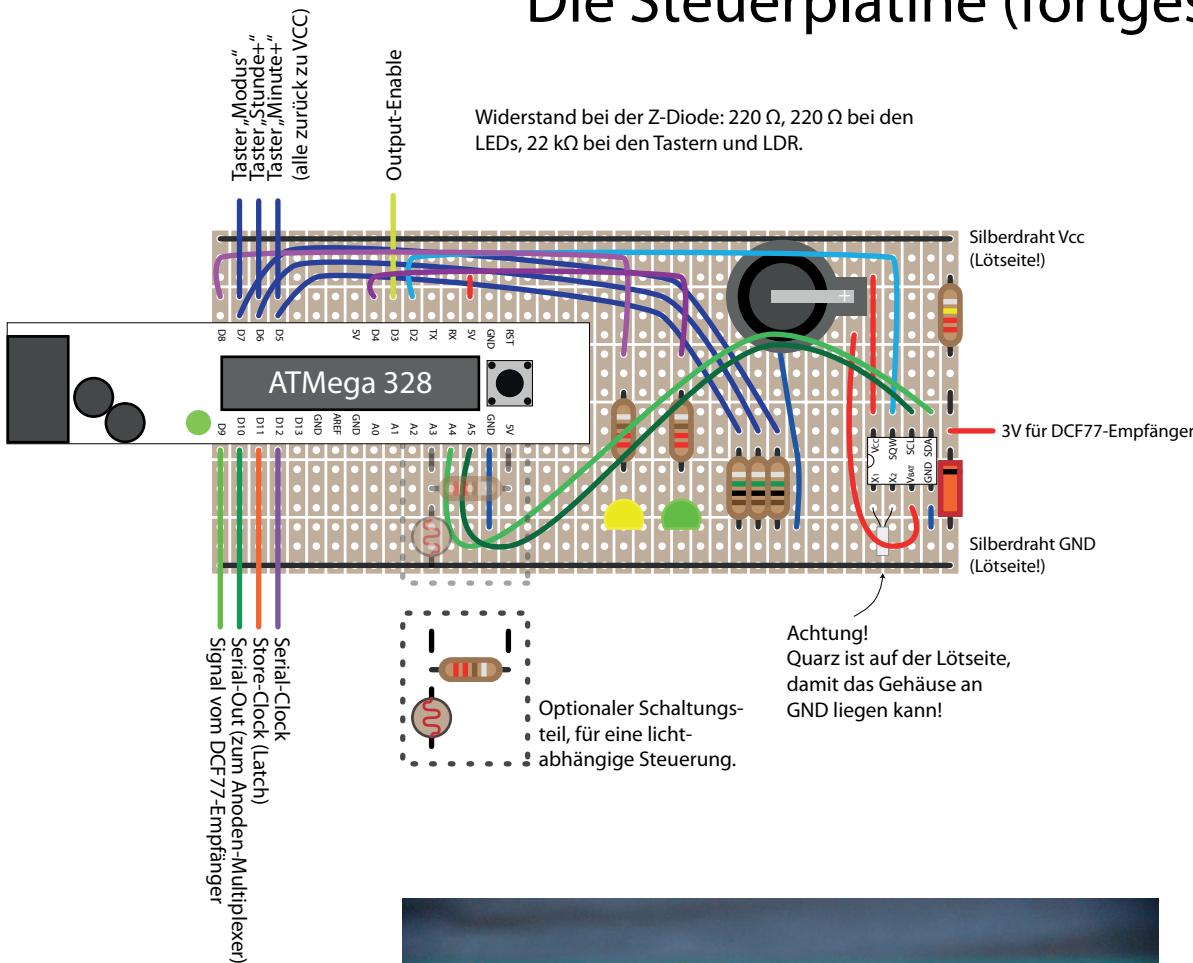


Die Steuerplatine ist das „Herz“ unserer QLOCKTWO und damit etwas aufwändiger. Um nicht durcheinander zu kommen, steckt man die Bauteile zur Probe mal in die Lochrasterplatine und markiert die Umrisse und wichtige Punkte mit einem Permanent-Marker.

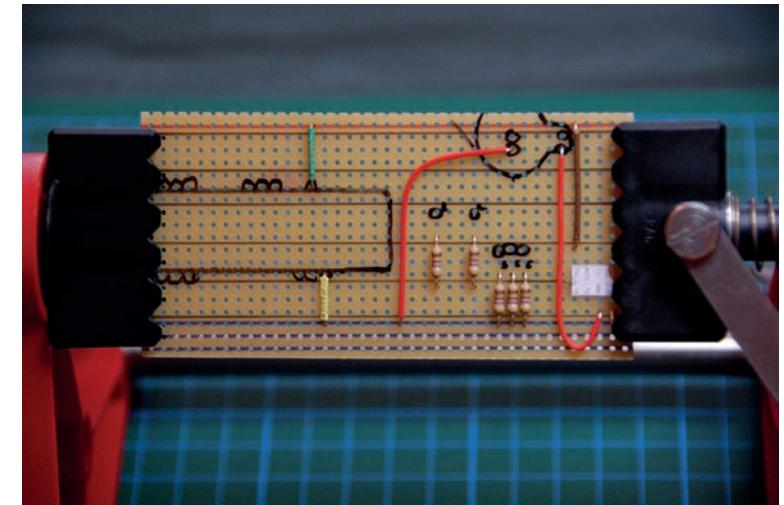
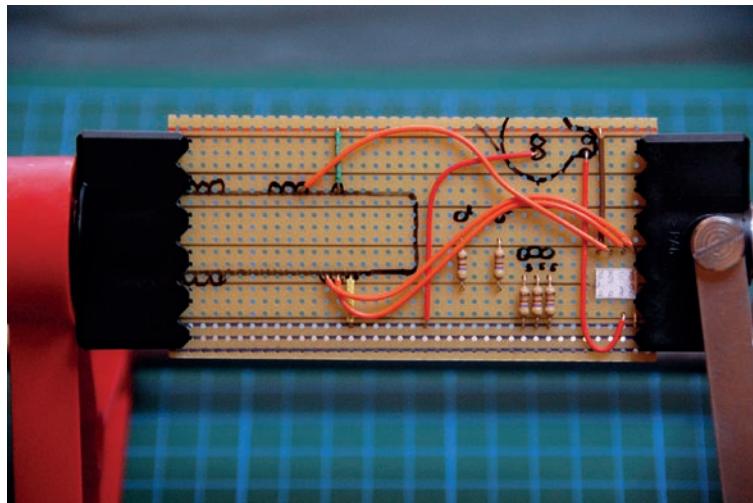


Achtung! Die Steuerplatine wird immer mal wieder verbessert. Auf den getrennten Download „Steuerplatine“ achten!

Die Steuerplatine (fortgesetzt)

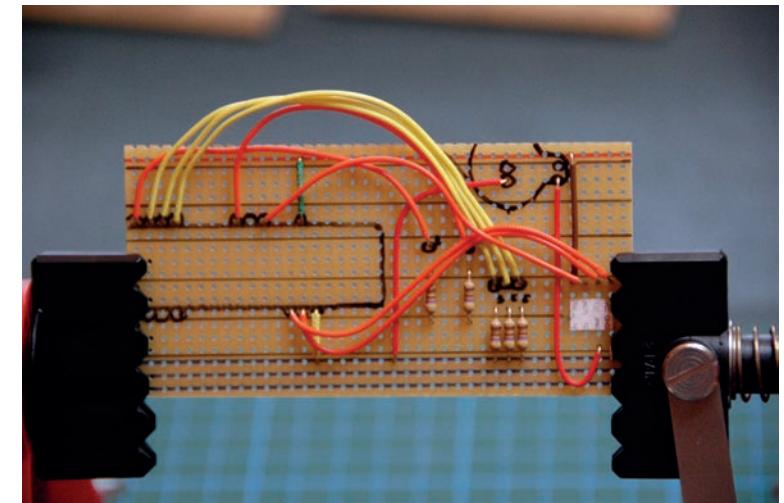


Rechts: Die SQW-Signalleitung (Square-Wave-Signal) und die Leitungen für den I²C-Bus werden angelötet (SCL = Serial Clock Line; SDA = Serial Data Line). Achtung! SCL und SDA gehen von der DS1307 vertauscht an das RBBB.

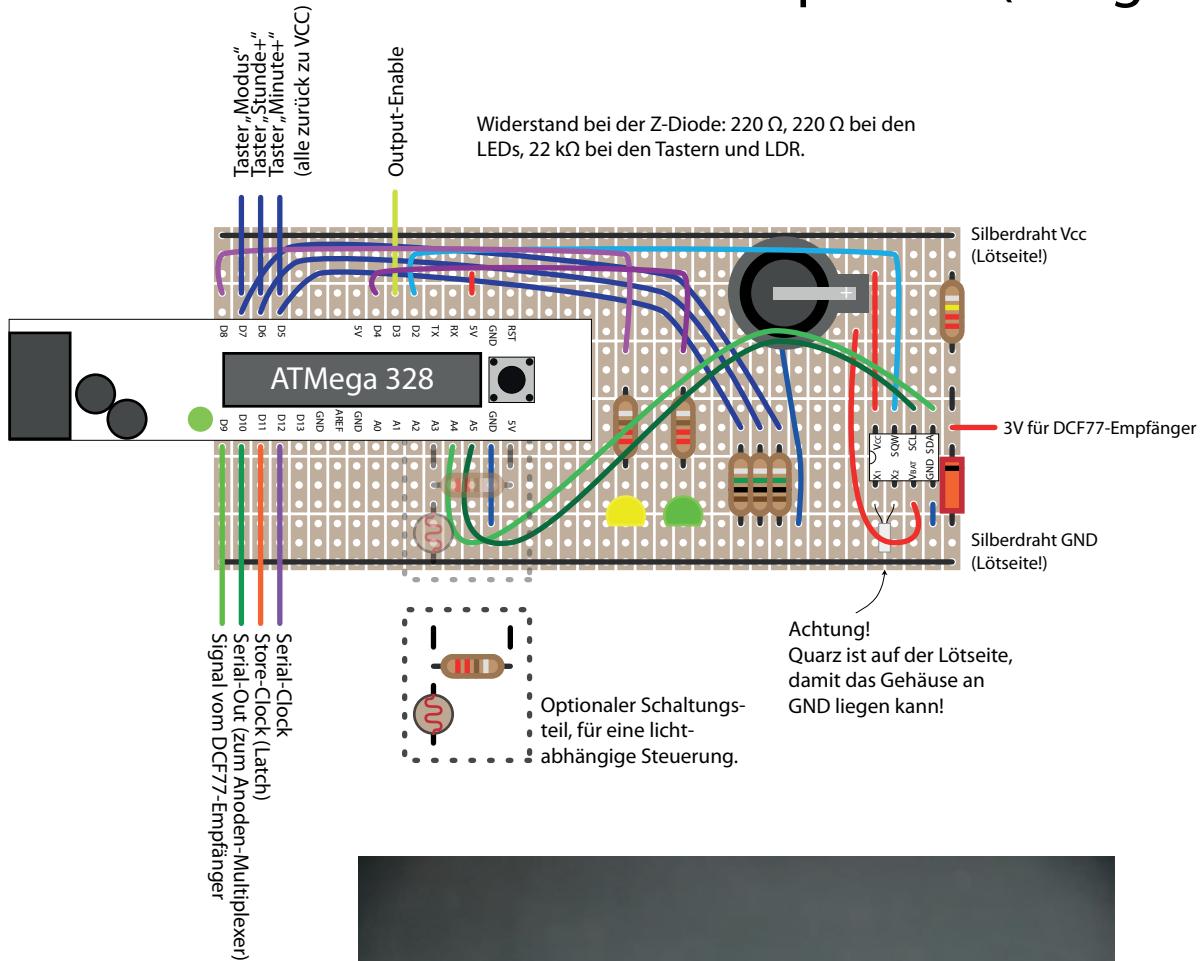


Oben: Zuerst werden Widerstände und Stromversorgungen angelötet.

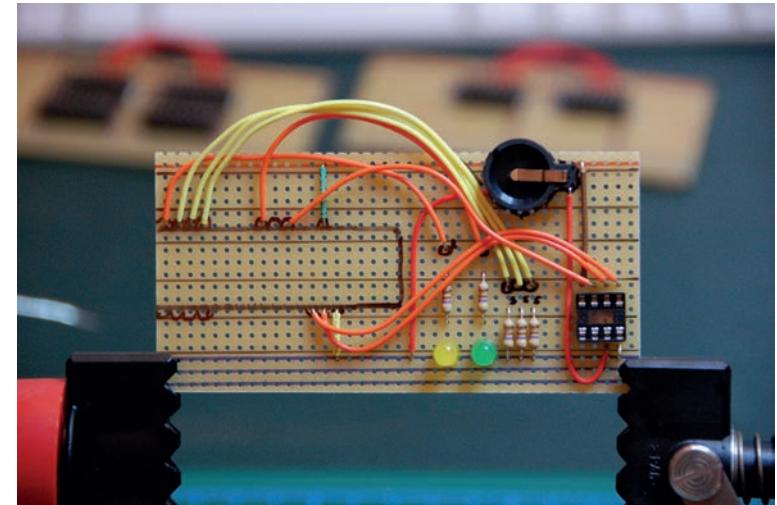
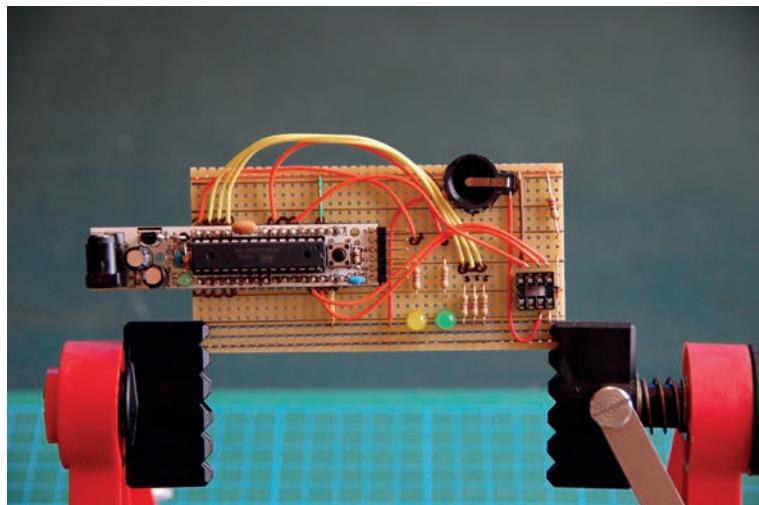
Unten: Die Leitungen zu den LEDs und Tastern werden angelötet.



Die Steuerplatine (fortgesetzt)

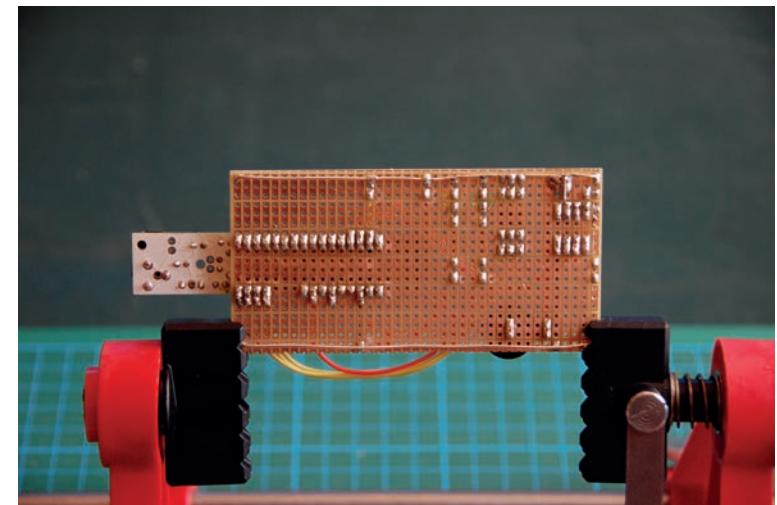


Rechts: Das RBBB wird eingelötet. Natürlich muß man es vorher zusammenlöten (nach der Anleitung bei JEE-Labs). Mit dem Power-Jack sollte man warten bis die Stromversorgung geklärt ist.

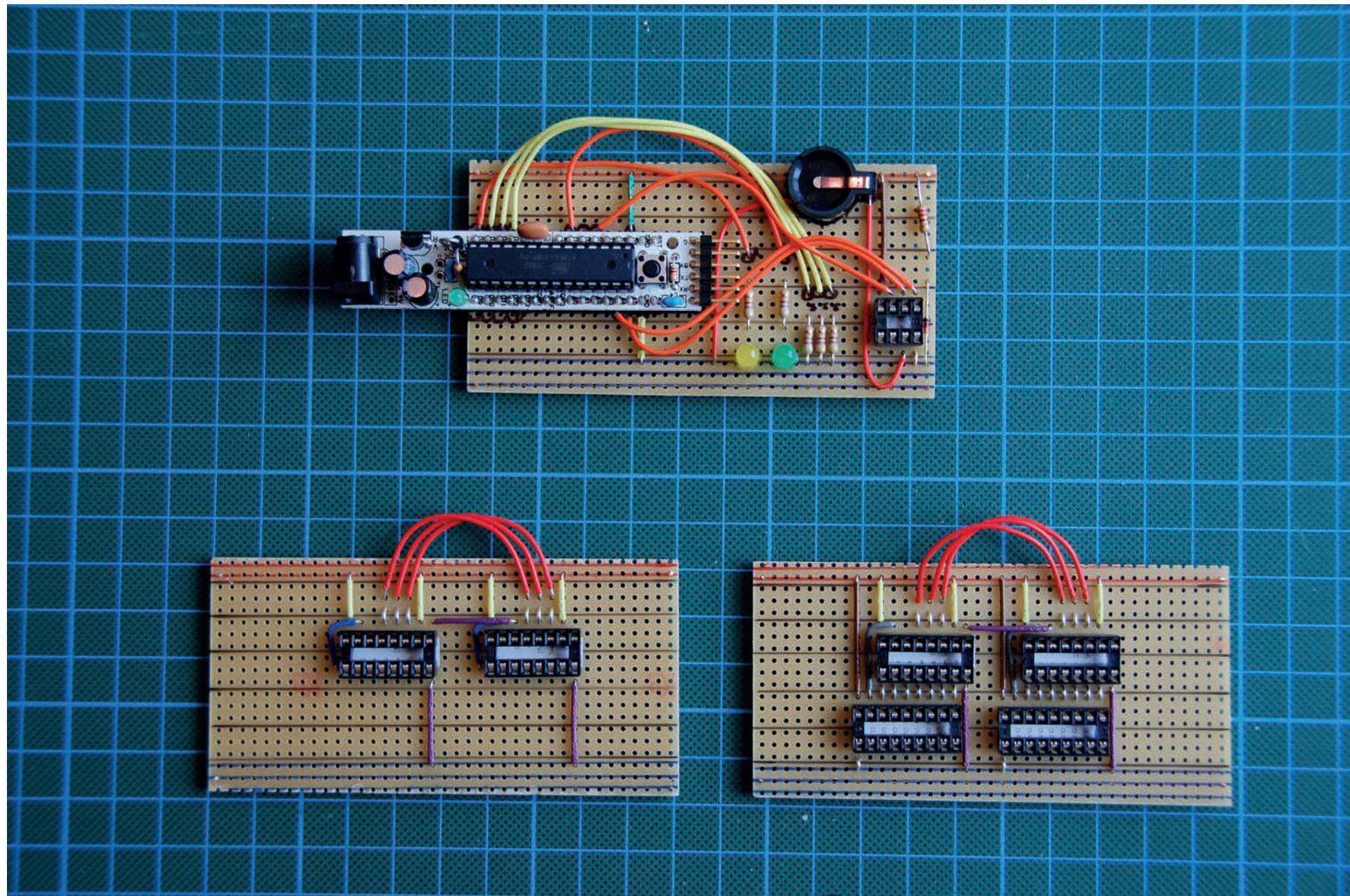


Oben: Die hohen Bauteile werden aufgelötet. Bei den LEDs auf die richtige Polung achten (flache Gehäuseseite zu GND)!

Unten: Rückseite mit Vcc und GND.



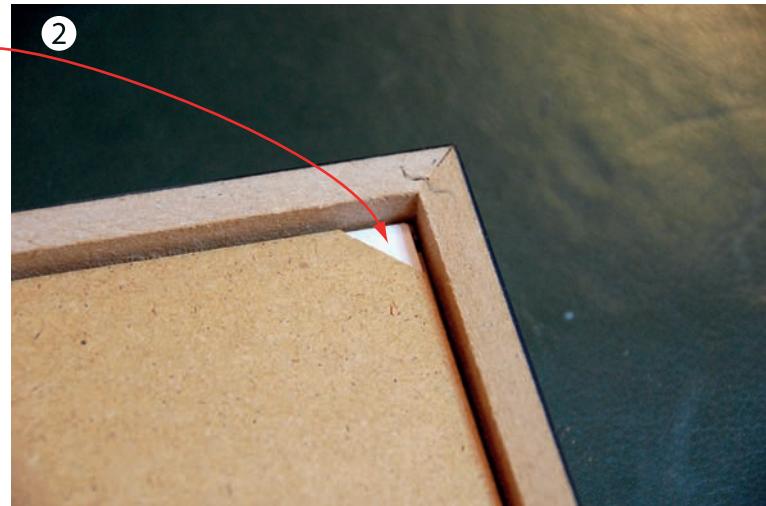
Ein erster Etappensieg!



Vorbereiten des Ribba-Rahmens



Der Ribba-Rahmen hat auf der Rückseite eine kleine Aussparung, mit derer Hilfe man die Rückwand herausnehmen kann.



Rückwand
Aufhänger
Schutzpapier
Außenrahmen
Innenrahmen
Passepartout



Als Erstes legen wir die Rückwand auf das Glas und dann den Innenrahmen darauf. Dann zeichnen wir die Innenkante des Innenrahmens auf der Vorder- und Rückseite der Rückwand nach. Diese Fläche ist „Sperrfläche“ auf der wir später nicht arbeiten können.

Vorbereiten des Ribba-Rahmens (fortgesetzt)

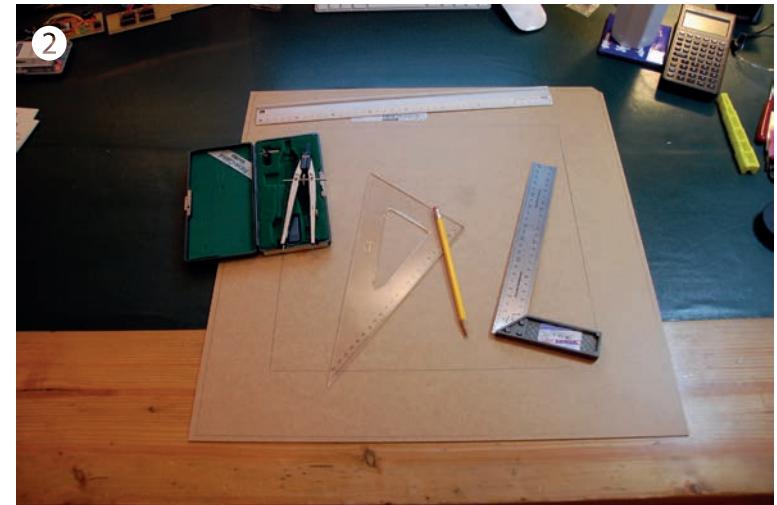


1

Jetzt werden wir viele Dinge auf die Rückwand zeichnen, also Bleistift spitzen und los!

Zuerst schreiben wir „Oben“ und „Unten“ mit einem Permanent-Marker auf beide Seiten.

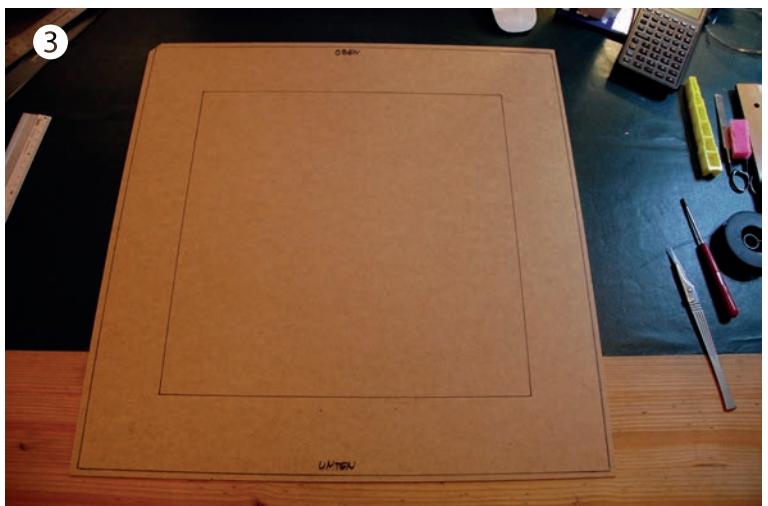
Dann zeichnen wir einen Innenrahmen mit einer Kantenlänge von 350 mm auf beide Seiten (Abstand zum Außenrand 75 mm).



2

Danach zeichnen wir das Gitter und die Bohrlöcher auf die Rückseite der Rückwand. Besser gesagt, auf irgendeine Seite, diese wird dadurch zu Rückseite. Wir haben 11 Spalten und 10 Reihen. Ein Kästchen hat damit eine Breite von $350 \text{ mm} / 11 = 31,82 \text{ mm}$ und eine Höhe von $350 \text{ mm} / 10 = 35 \text{ mm}$. Diese Strecken kann man gut mit dem Zirkel aufnehmen und dann an dem Rand des Innenrahmens abtragen. Dann werden Diagonalen in die Kästchen gezeichnet. Der Schnittpunkt der Diagonalen ist die Vorgabe für das Bohrloch.

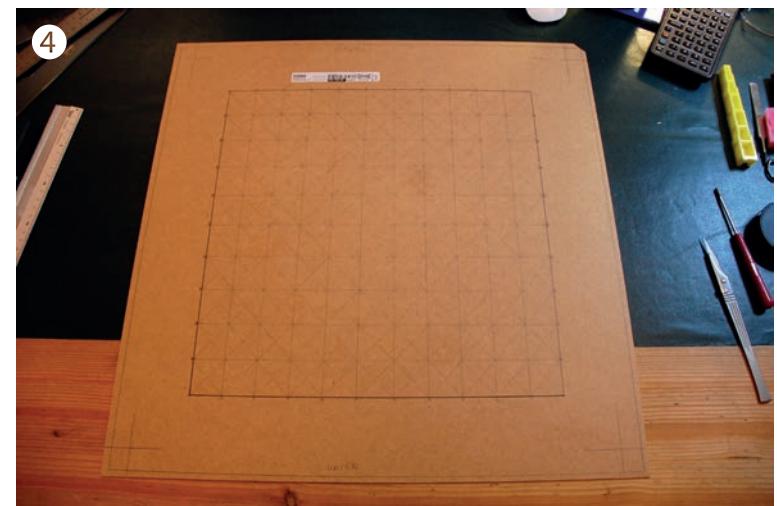
Natürlich kann man sich auch die Bohrschablone ausdrucken und damit die Löcher markieren. ☺ Pro-Tipp von Michael: Wer keinen Posterplotter hat, kann die Vorlage via Adobes Acrobat Reader in 1:1 „gekachelt“ ausdrucken und zusammenkleben.



3

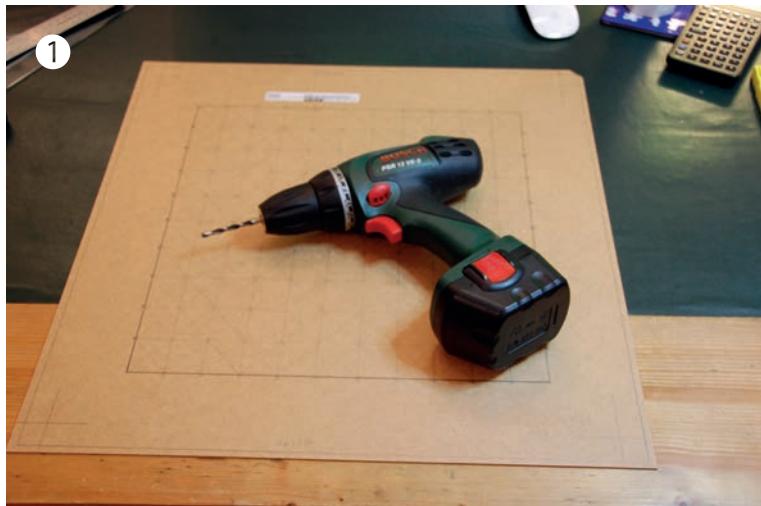
Die Lochmarkierungen für die Eck-LEDs haben einen Abstand von 25 mm zum Rückwand-Rand.

Alles nochmal überprüfen, wenn die Löcher erst gebohrt sind, ist es zu spät.



4

Löcher bohren



Oben: Für die LEDs bohrt man Löcher mit 5 mm Durchmesser.

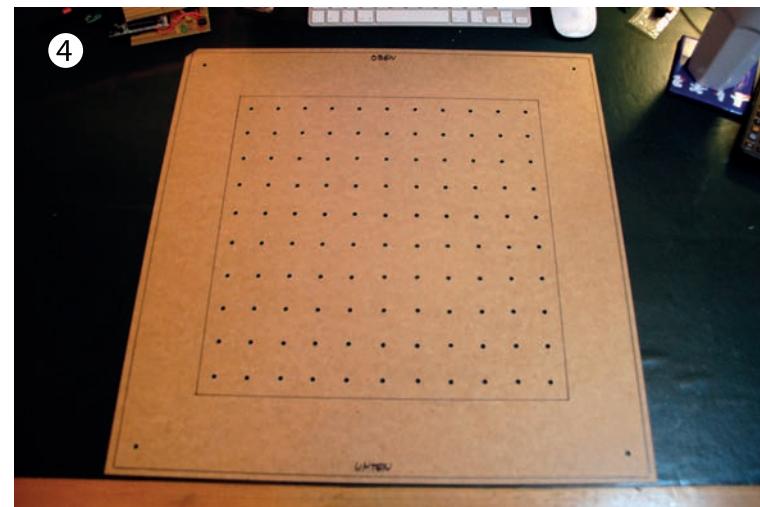
Unten: Für die Silberdrahdurchführungen nimmt man einen sehr feinen Bohrer.



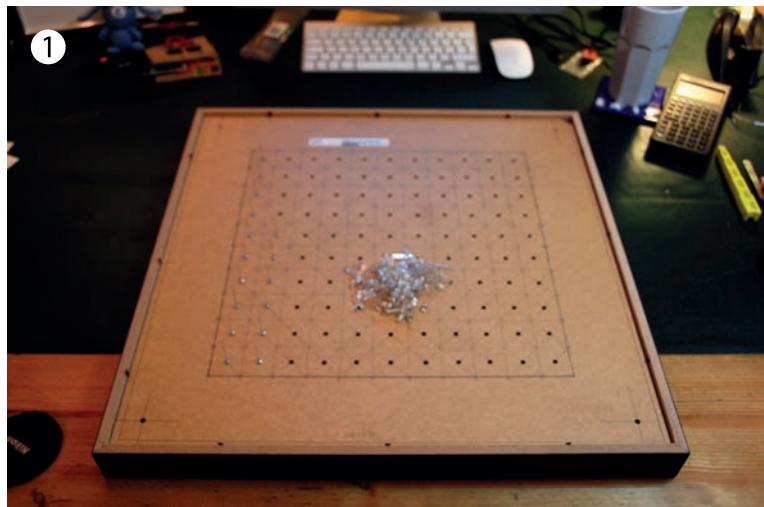
Achtung! Der Ribba-Rahmen ist nicht aus Holz, sondern aus gepresster Pappe. Daher beim Bohren mit wenig Druck arbeiten.



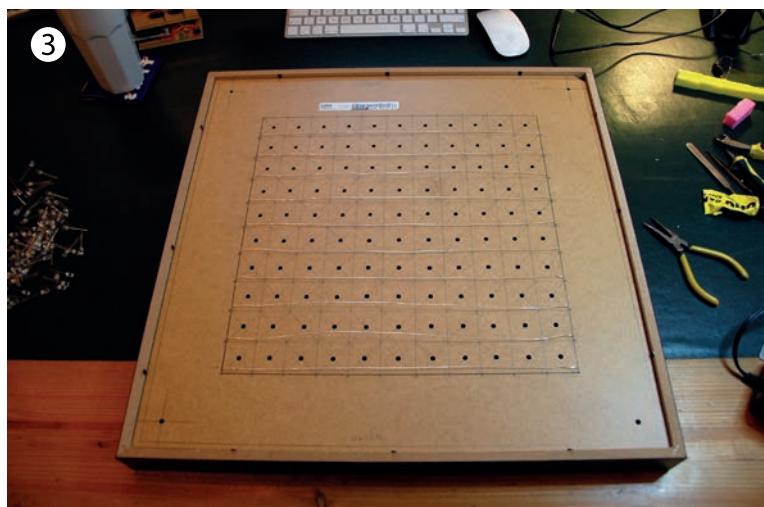
Rechts: Fertig gebohrte Rückwand der QLOCKTWO.



LEDs bestücken und Matrix löten



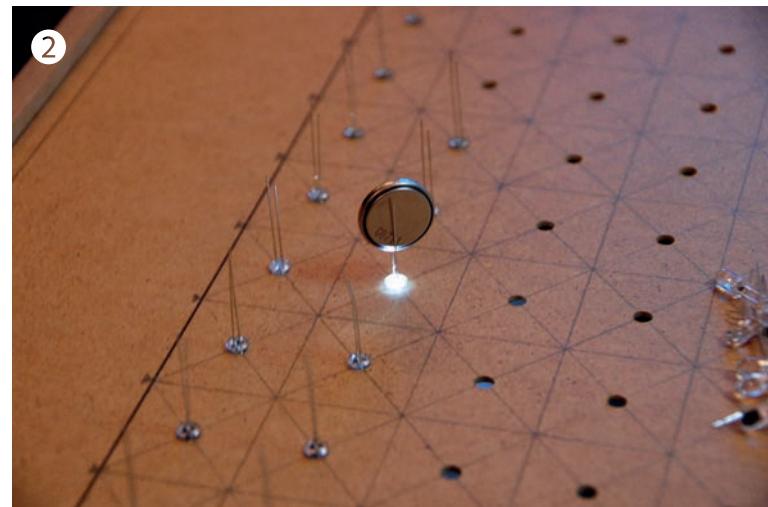
Unten: Zuerst schneiden wir 10 Silberdrähte ab und stecken diese durch die Löcher, so daß wir 10 Querreihen haben. Die Silberdrähte sollten etwa 2 cm auf der anderen Seite überstehen.



Die weißen LEDs werden nach dem LED-Löt-Schema auf ein Gitter von Silberdrähten gelötet (Kathode unten rechts, Anode oben links). Auf gute Lötstellen achten! Die Verbindung zwischen LED-Bein und Silberdraht dient auch der Wärmeabfuhr, das ist für die Lebensdauer der LEDs wichtig.

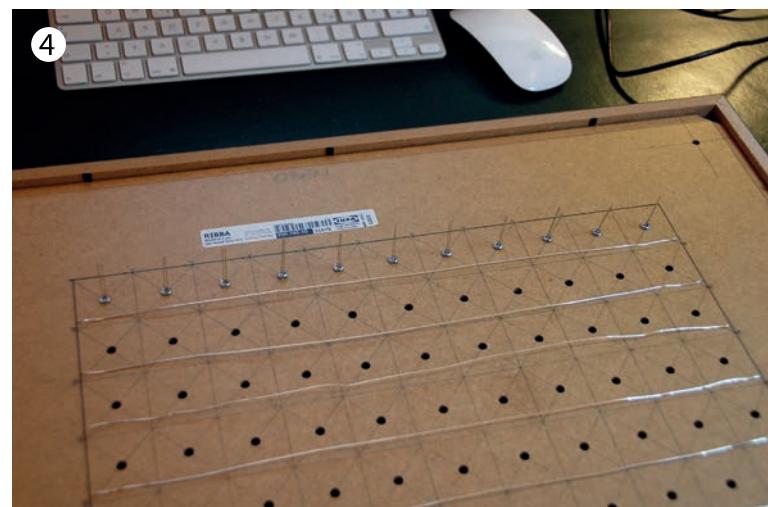


Rechts: Dann bestücken wir die erste Reihe oben mit LEDs. Wenn man gleich die ganze Matrix bestückt, erschwert das die Lötarbeit.

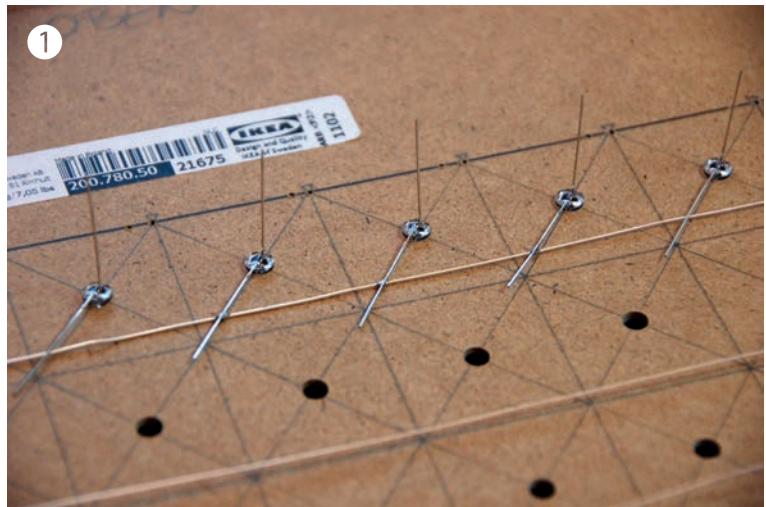


Oben: Die Polarität und Funktionstüchtigkeit einer LED kann man mit einer alten CR 2032-Knopfzelle testen. Nicht mit anderen Batterien (oder nur mit Vorwiderstand).

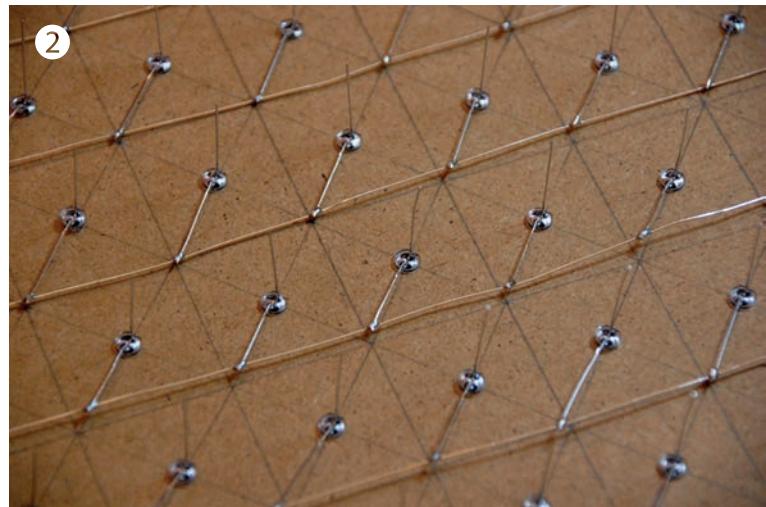
Pro-Tipp: Eine CR 2032 kostet 0,33 € bei Reichelt und 4,50 € im Elektrofachhandel.



LEDs bestücken und Matrix löten (fortgesetzt)

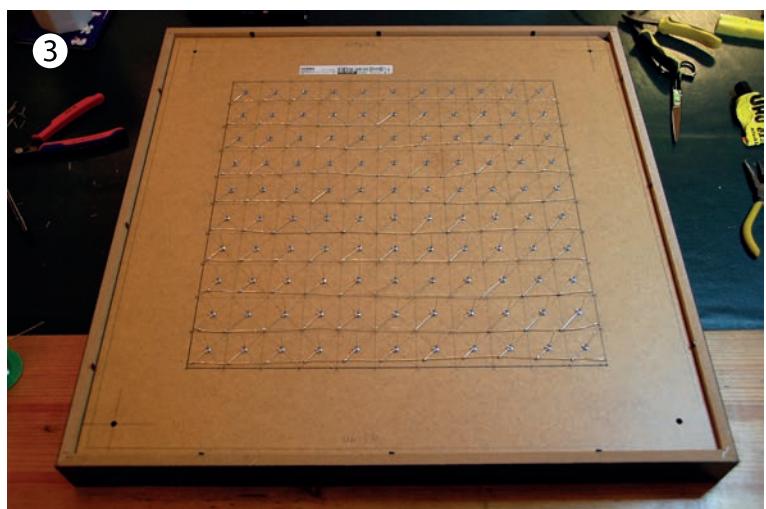


Links: Die Anoden-Beine der LEDs werden umgebogen (die anderen nicht!) und dann auf den Silberdraht gelötet. Der Silberdraht dehnt sich beim Löten aus, daher fangen wir in der Mitte an! Wieder ist eine zweite Person hilfreich, die die Beine runterdrückt.

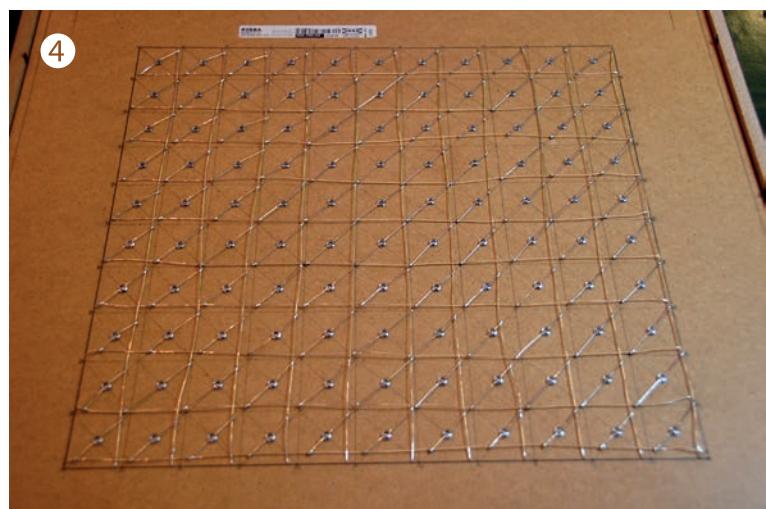


Oben: Der Überstand der Beine wird abgezwicke.

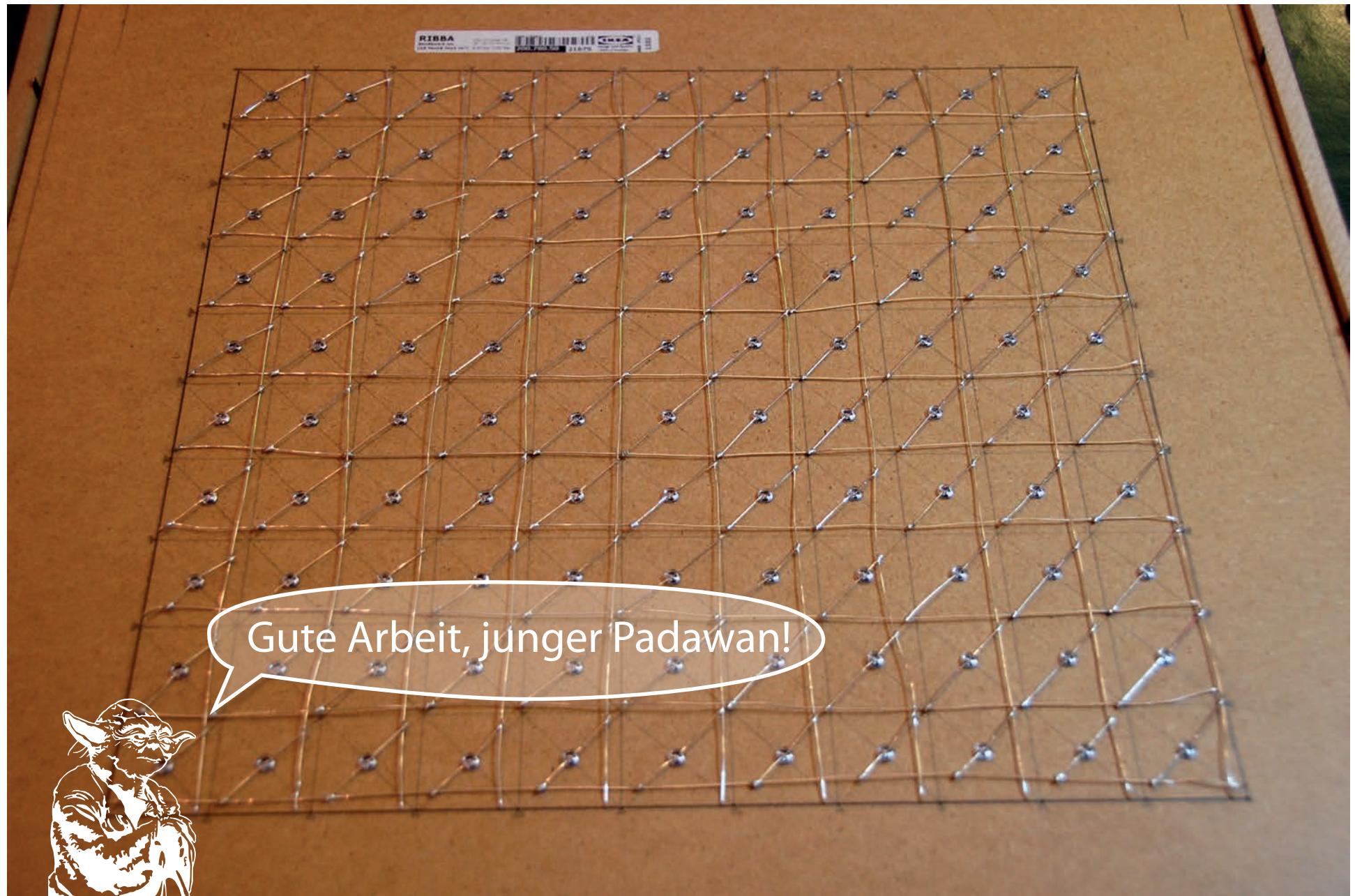
Unten: So sollte die erste Hälfte der Matrix aussehen. Da das eine ermüdende Arbeit ist, muß man in dieser Phase besonders auf die Polung der LEDs und sauberes Löten achten.



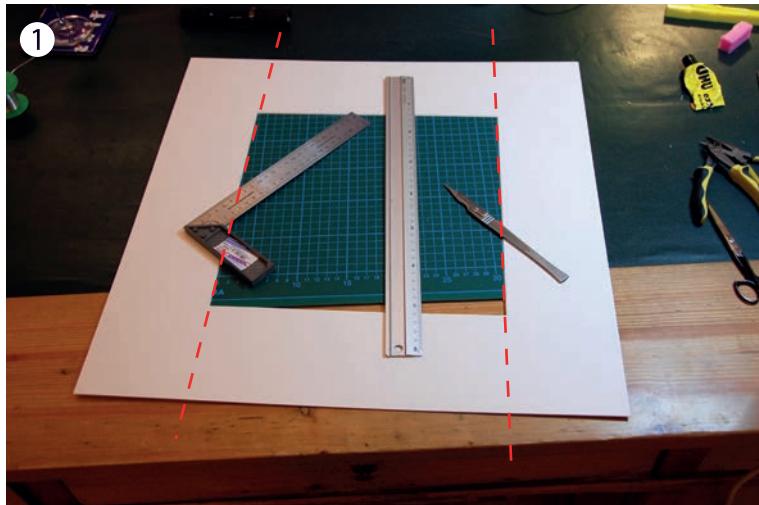
Rechts: Dann werden die Silberdrähte für die Kathoden eingezogen und die Kathoden der LEDs analog zu den Anoden angelötet.



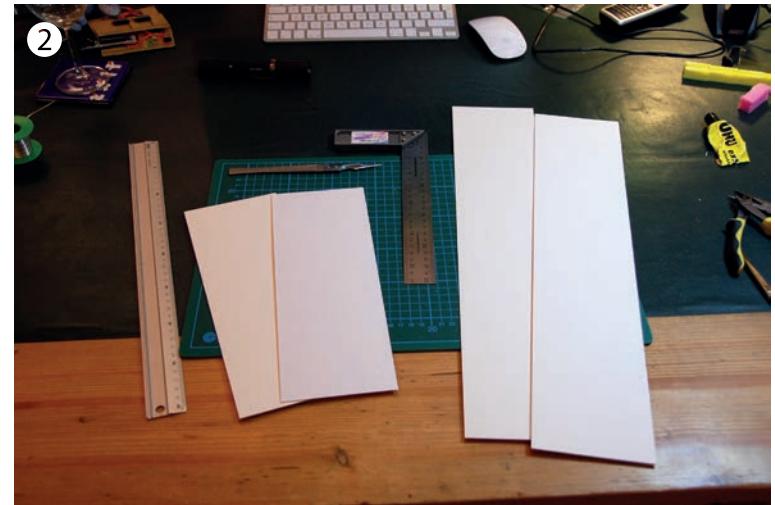
LEDs bestücken und Matrix löten (fortgesetzt)



Matrix isolieren

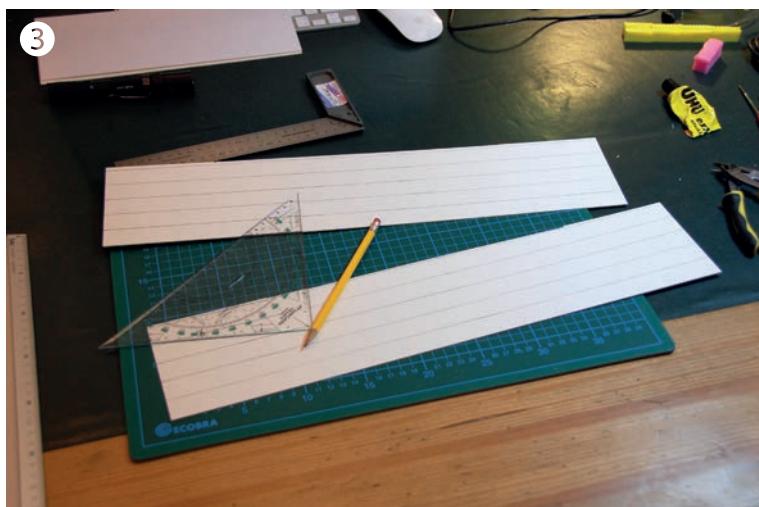


Links: Für die Isolierung zwischen den Anoden und Kathoden zerlegen wir das Passepartout von dem Ribba-Rahmen.

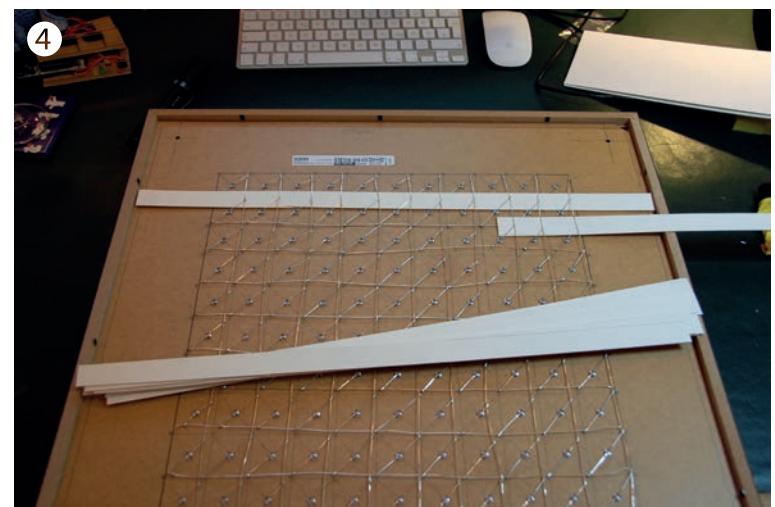


Oben: Wir brauchen zehn Streifen in der Länge der Rahmenbreite.

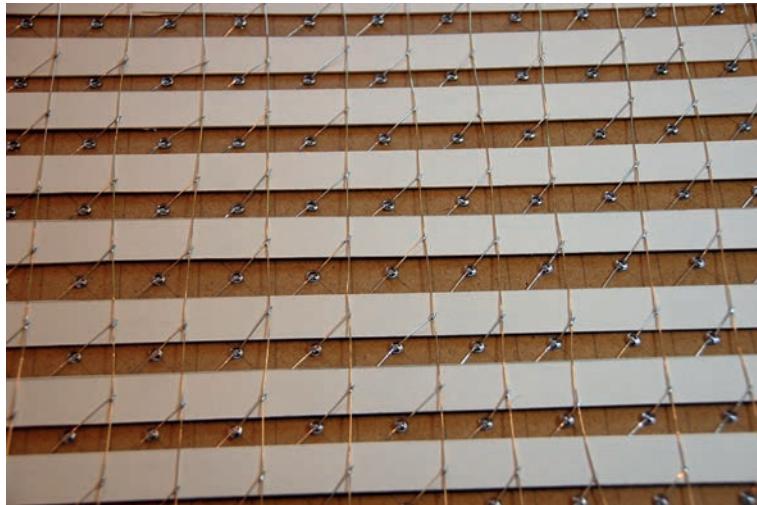
Unten: Wir unterteilen die langen Stücke in 2 cm-Streifen. Das Schneiden der Streifen geht gut mit einem Teppichmesser.



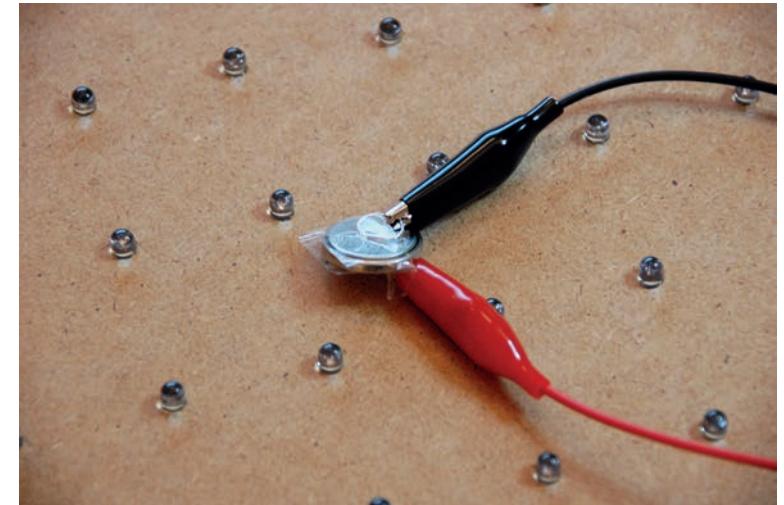
Rechts: Die Streifen werden dann vorsichtig zwischen die Anodenreihen (Oberseite) und Kathodenreihen (Unterseite) geschoben. Wenn sich dabei Beine von den Silberdrähten lösen: Super! Wir haben eine kalte Lötstelle gefunden, die uns nachher geärgert hätte. Einfach nachlöten.



Matrix testen

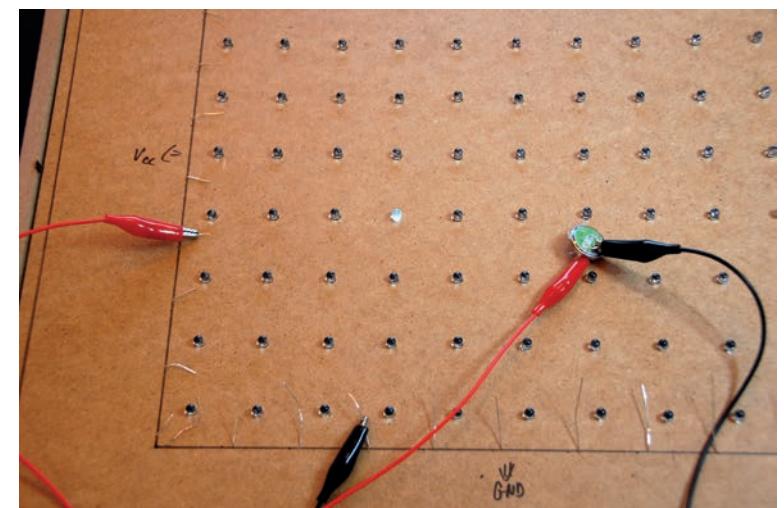
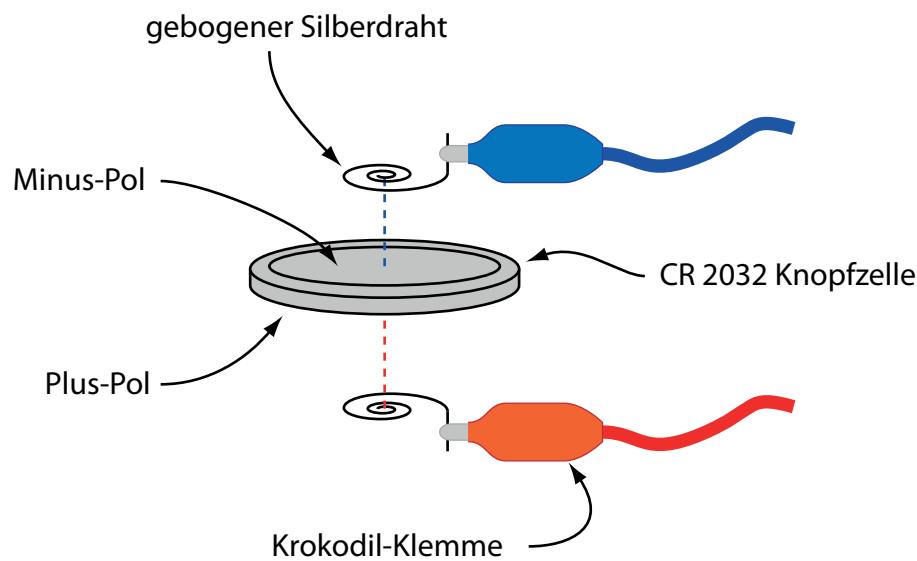


Links: Nachdem die Matrix isoliert ist, geht es an das Testen. Dazu versorgen wir jede Reihe und Spalte einmal kurz mit Strom und schauen, ob die entsprechende LED leuchtet.

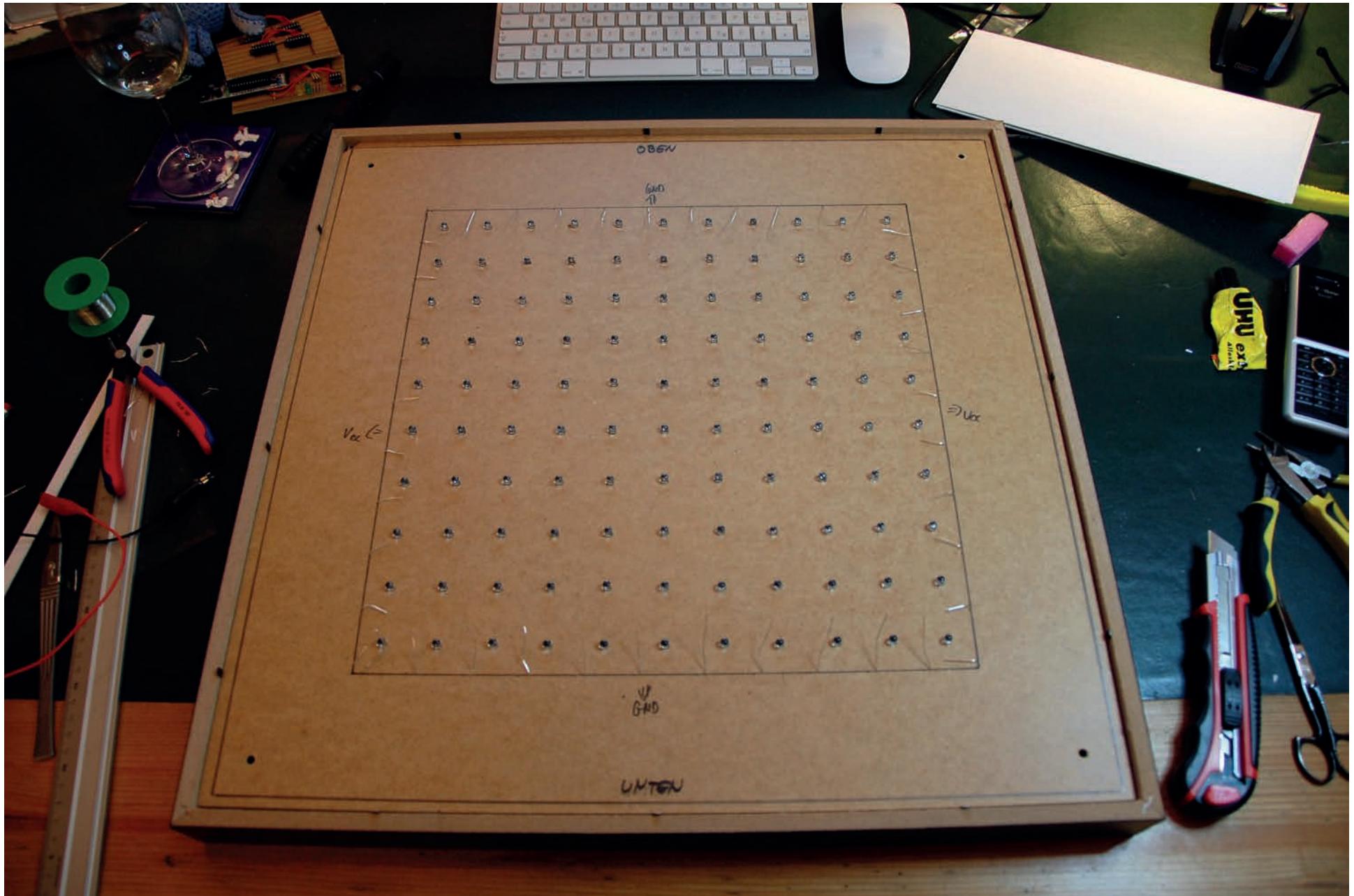


Oben: Für unser high-tech Matrix-LED-Testgerät biegen wir zwei Silberdrahtstücke zu Spiralen, kleben sie mit Tesa an eine alte CR 2032-Knopfzelle und klemmen Krokodil-Klemmen an.

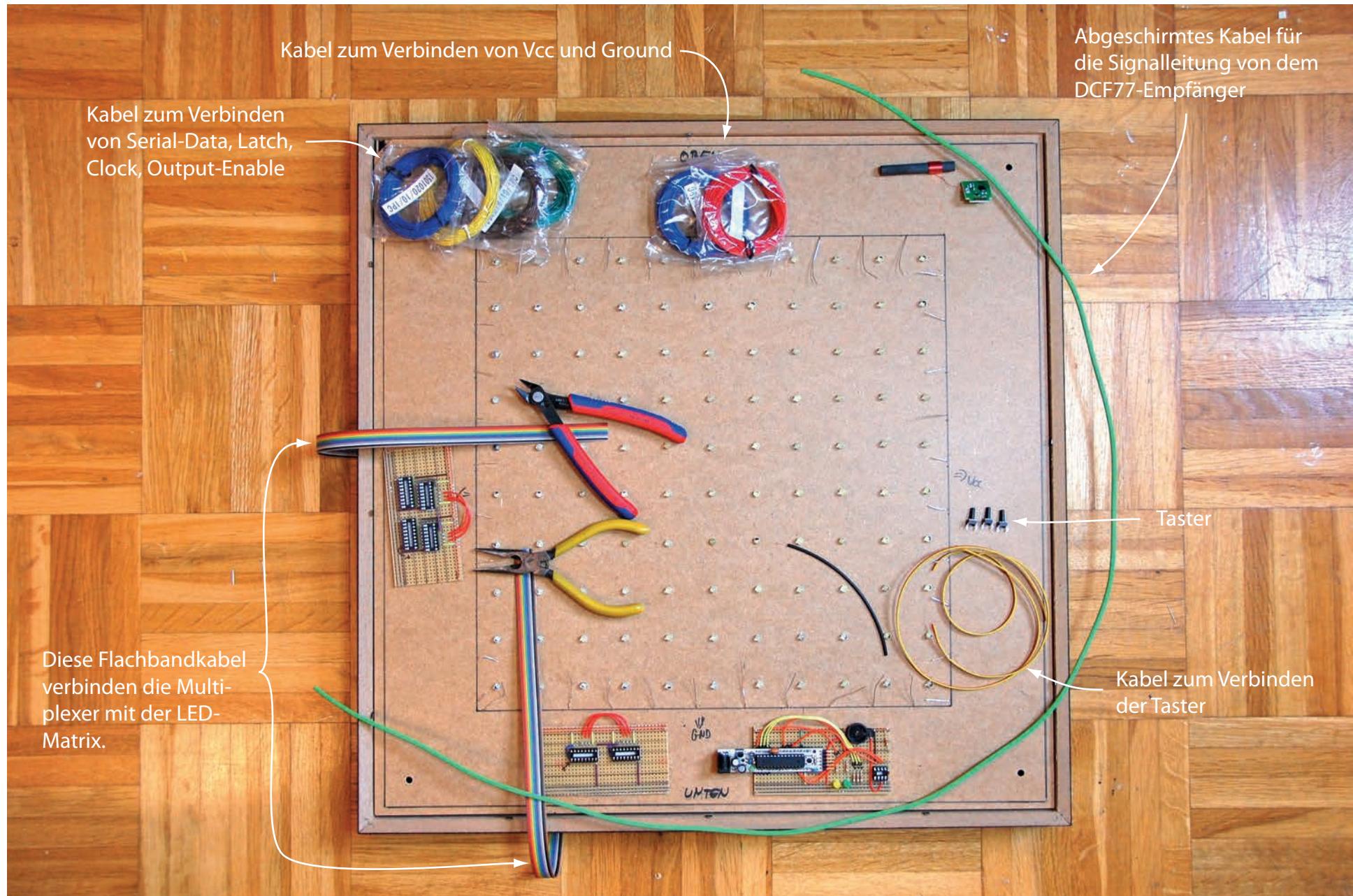
Unten: Die angesprochene LED leuchtet wie erwartet.



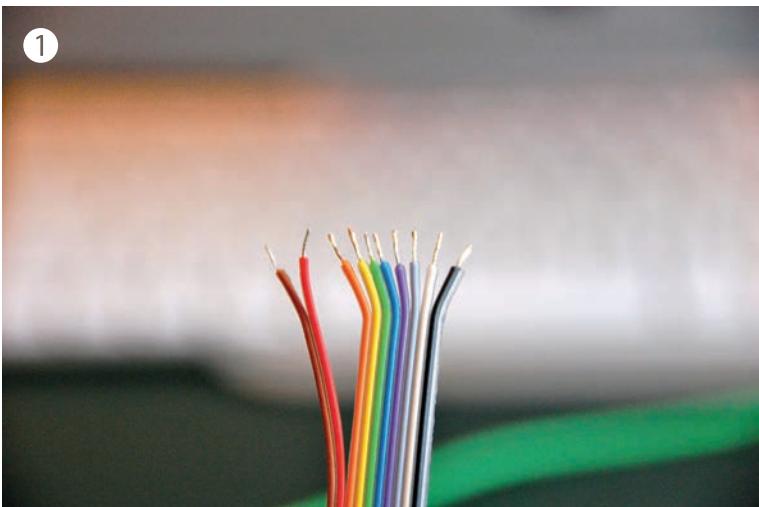
Ein weiterer Etappensieg!



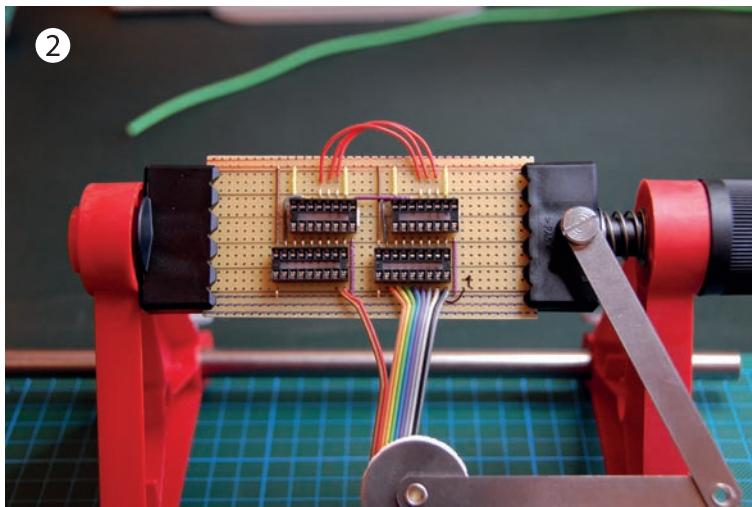
Steuerplatinen anschliessen



Steuerplatinen anschliessen (fortgesetzt)

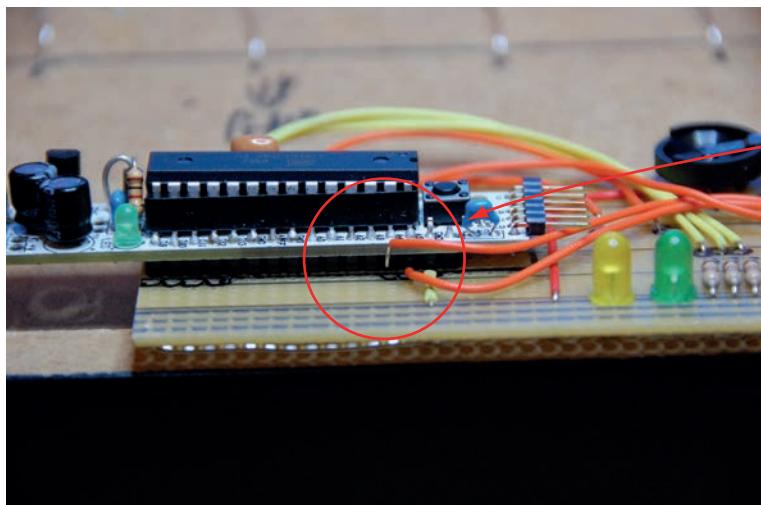


Die Flachbandkabel (oder auch Einzelkabel) werden abisoliert und verzинnt. Achtung! Wir brauchen einmal 10 Adern und einmal 11 Adern!



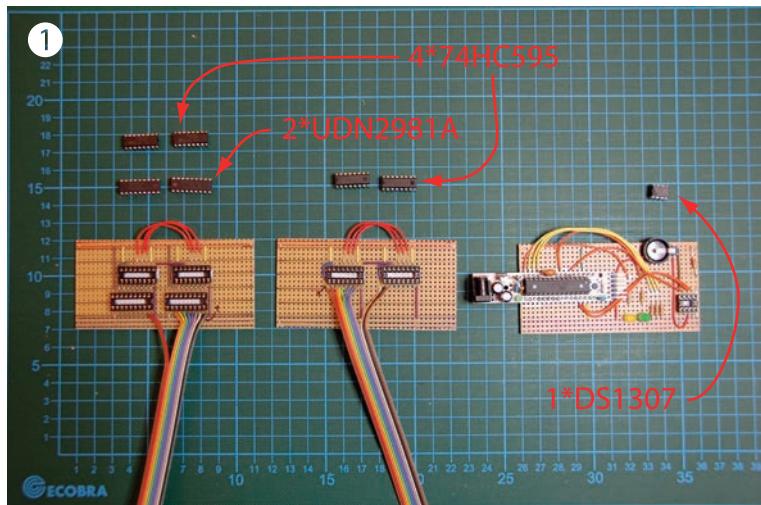
Dann werden sie an die Ausgänge vom Kathoden- und Anodenmultiplexer gelötet.

kleiner Einschub am Rande:



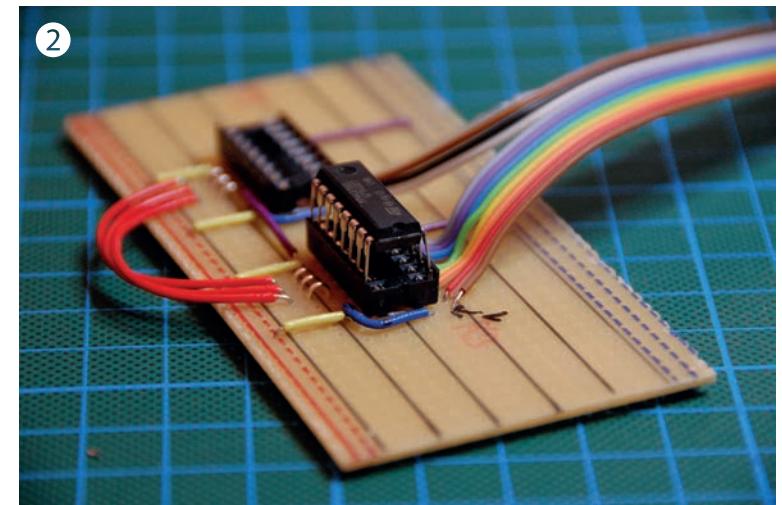
Wie schon vorher erwähnt: Immer, wenn irgendwo bei der Arbeit Kabel wegfallen oder sich lösen: Großartig! Der Fehler hätte uns später geärgert (und da kommen erfahrungsgemäß genug). Nicht ärgern sondern freuen, beseitigen und weiter gehts!

Chips bestücken



Links: Wenn die Flachbandkabel angelötet sind, können die Chips eingesetzt werden. Ein großer Moment im Leben eines QLOCKTWO-Bastlers! (Achtung! Die UDN2981A-Chips sind um 180° zu den Shift-Registern gedreht!)

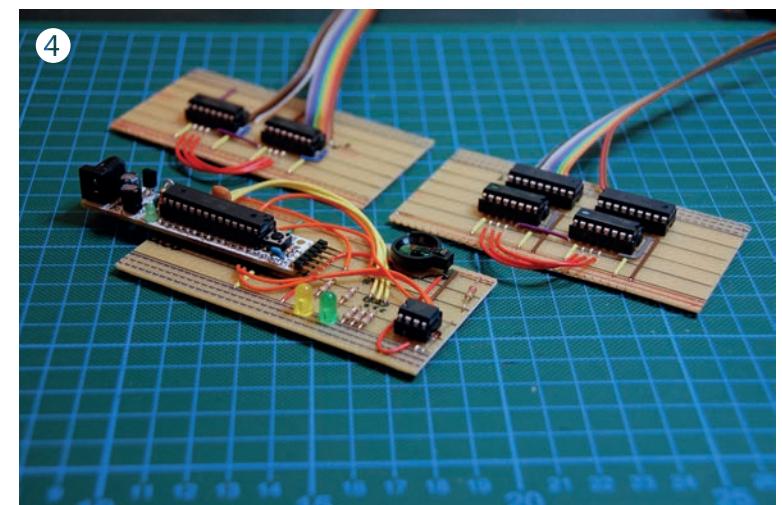
Rechts: Die Chips sind ein klein wenig größer als die Fassungen. Daher muß man sehr behutsam die Beine etwas zusammendrücken.



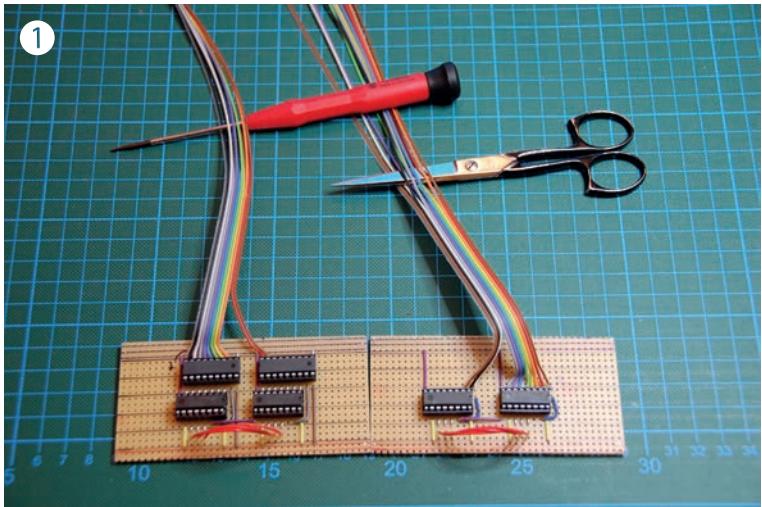
Unten: Die Chips haben als Markierung für die Richtung eine kleine Aussparung oder „Nase“. Die Fassungen haben diese Nase auch.



Unten: Jetzt sind alle Platinen bereit und müssen „nur noch“ an die Matrix angeschlossen und untereinander verbunden werden.

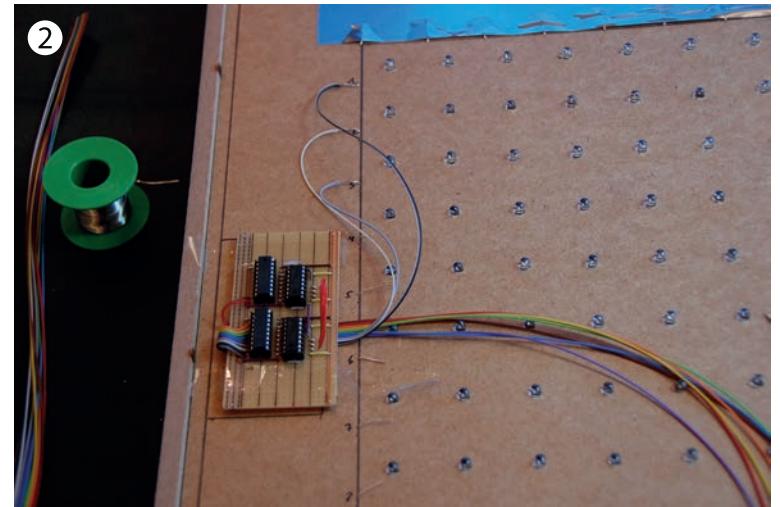


Steuerplatinen anschliessen (fortgesetzt)



1

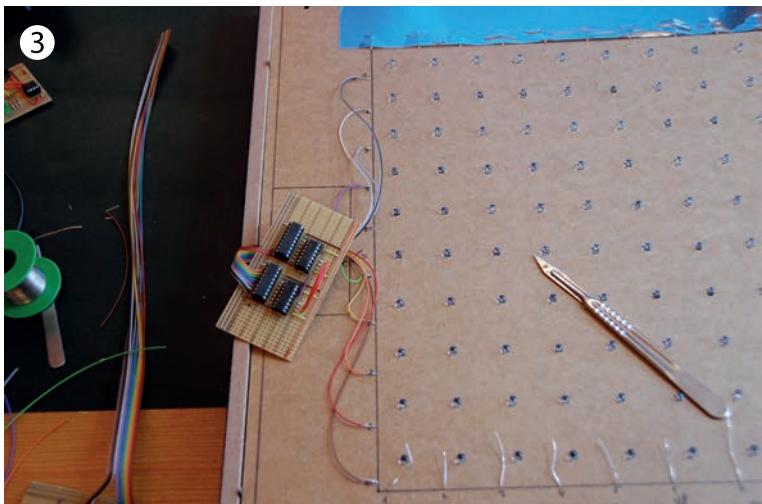
Links: Die Flachbandkabel werden am Besten mit einem Skalpell in die Einzeladern aufgetrennt. Ein paar Zentimeter kann man zusammen lassen, da das Kabel unter der Platine geführt wird.



2

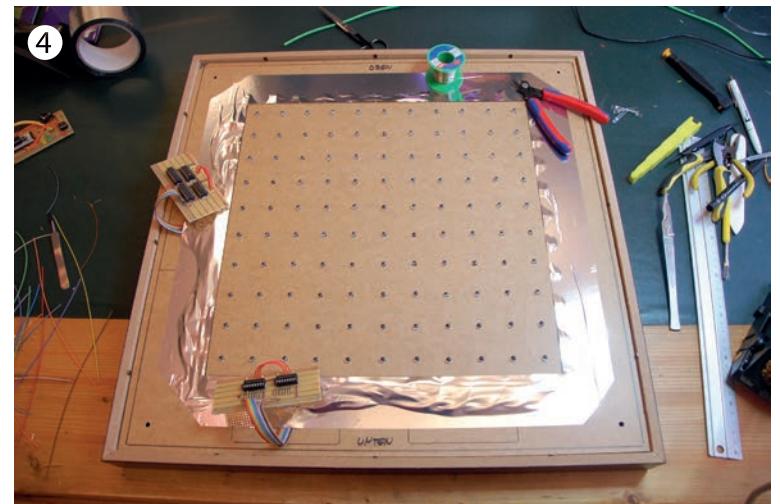
Oben: Die Kabel werden an die Matrix angelötet. Achtung! Der mit „1“ markierte Ausgang ist die oberste Reihe, das bedeutet, die Kabel müssen beim Anoden-Multiplexer gekreuzt werden.

Unten: Der fertig angelötete Anoden-Multiplexer.



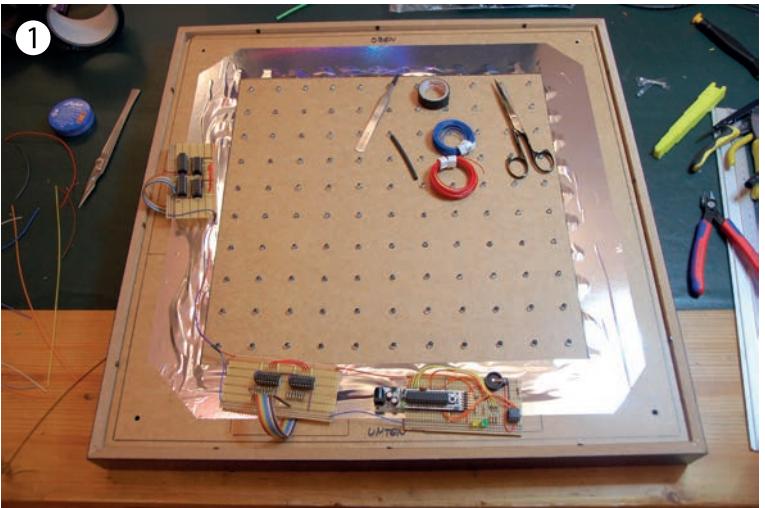
3

Rechts: Der zweite Multiplexer wird genauso angelötet, die Adern müssen aber nicht gekreuzt werden. Danach werden die Ausgänge der Matrix isoliert, damit es keine Kurzschlüsse gibt und die Kabel sich nicht verheddern.



4

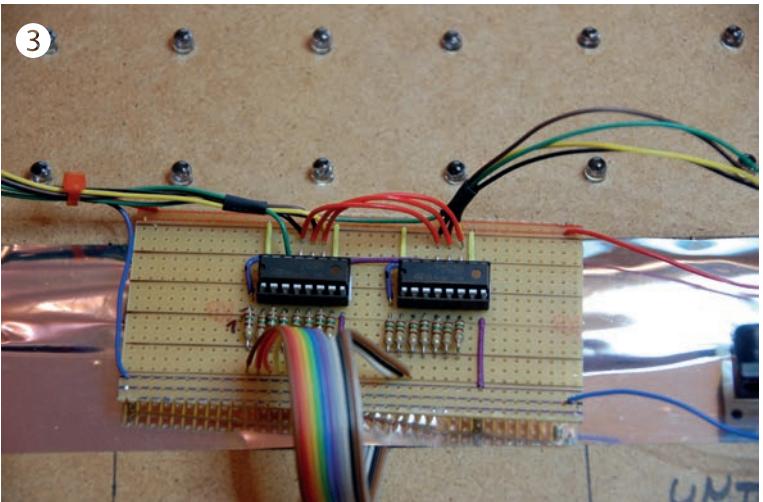
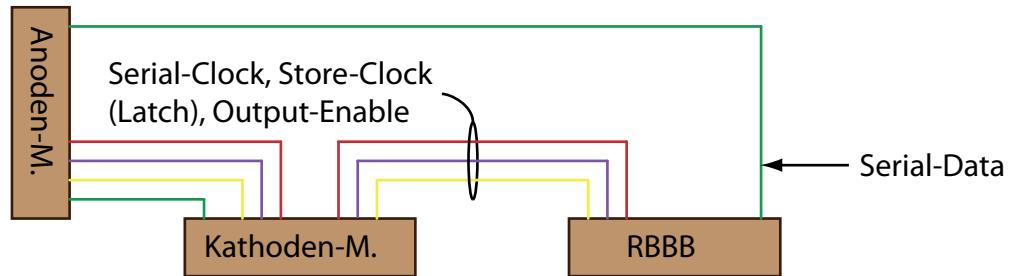
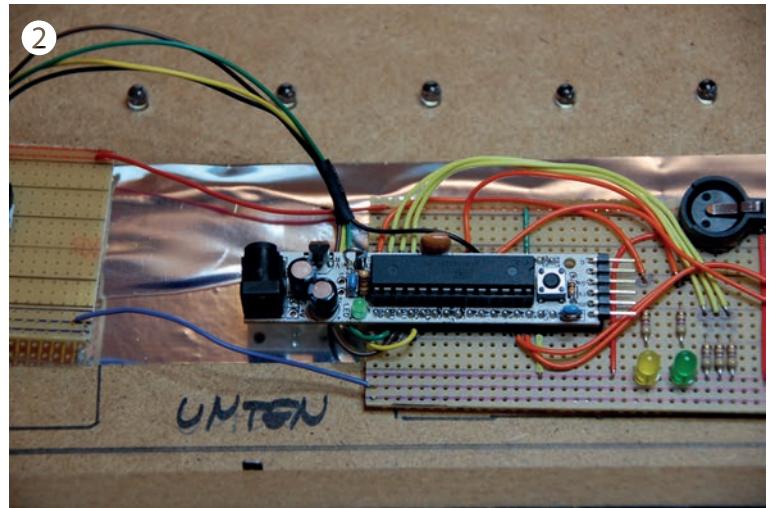
Steuerplatinen verbinden



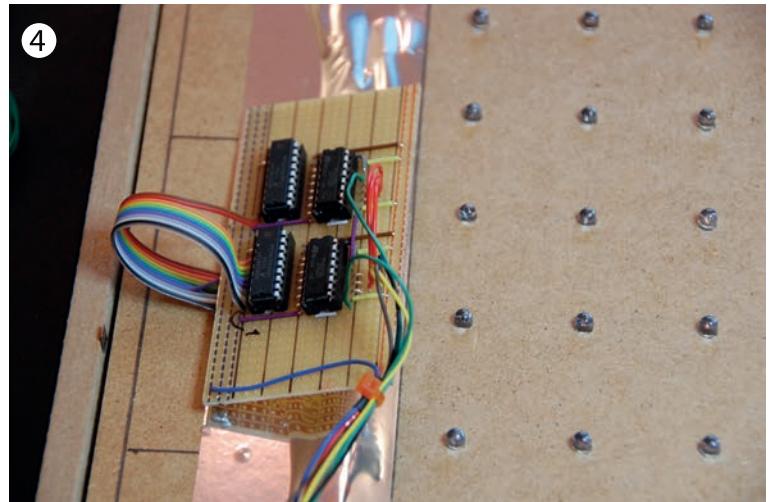
Unten: Serial-Clock, Store-Clock (Latch) und Output-Enable gehen in den Kathodenmultiplexer rechts rein und links raus. Das Serial-Data-Kabel geht weiter, links kommt es vom Anoden-Multiplexer.

Links: Mit roten und blauen Kabeln werden die Vcc- und Ground-Bereiche auf den Platten miteinander verbunden.

Rechts: Serial-Clock, Store-Clock (Latch), Serial-Data und Output-Enable werden aus dem RBBB herausgeführt. Das Serial-Out-Kabel muß so lange sein, daß es an den Anoden-Multiplexer oben passt! Der Rest der Kabel wird von Platine zu Platine durchgeschliffen.



Rechts: Serial-Clock, Store-Clock (Latch) und Output-Enable kommen vom Kathoden-Multiplexer. Serial-Data geht vom RBBB kommend oben rein, unten geht es raus und wird an den Kathoden-Multiplexer gereicht.

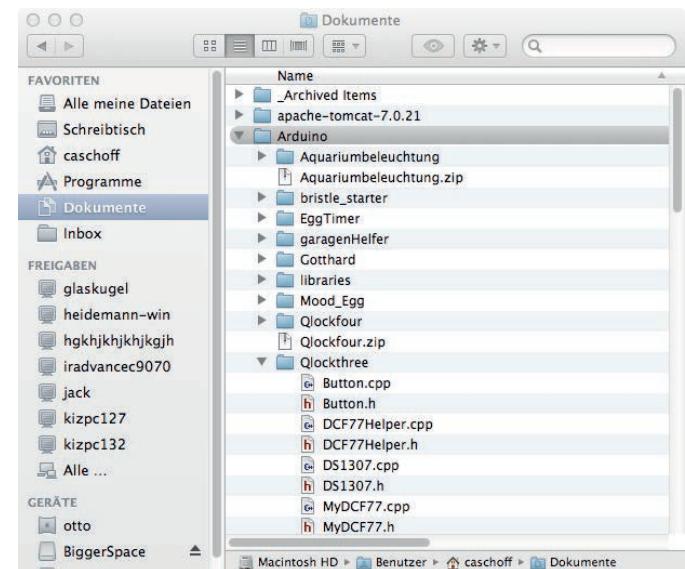


Ausflug: Wie kommt die Software auf den RBBB?

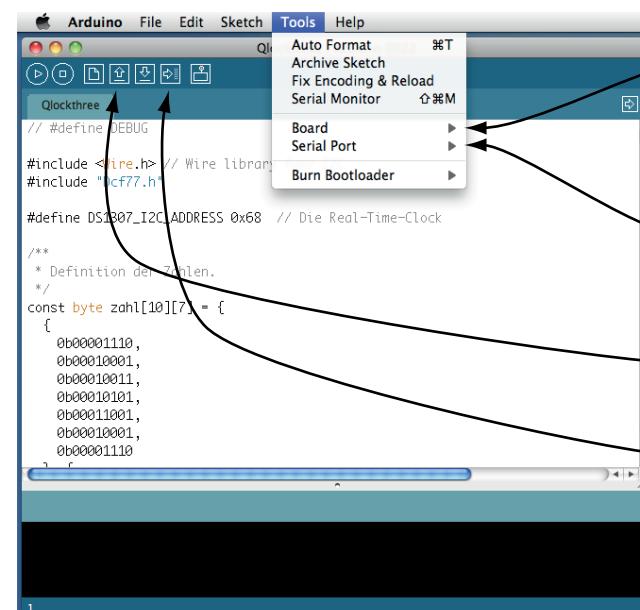
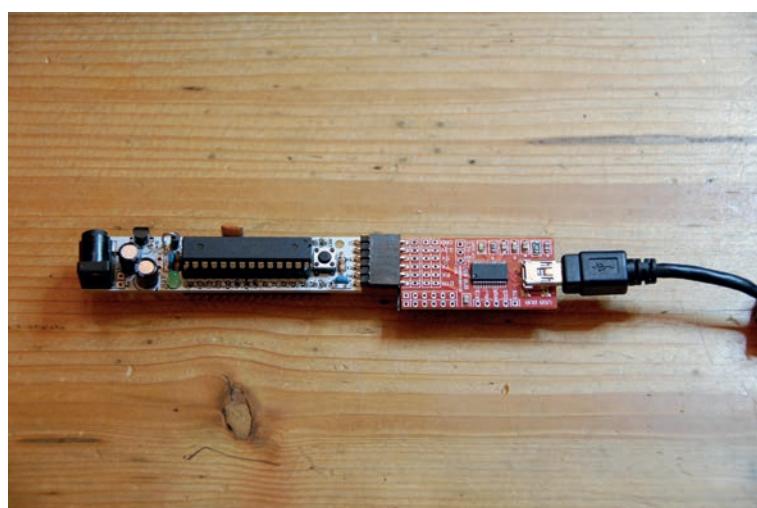


Links: Die Entwicklungsumgebung lädt man unter www.arduino.cc kostenlos herunter und installiert sie (zusammen mit den FTDI-Treibern!).

Rechts: Der entpackte Arduino-Ordner **muß** unter Documents (Windows: „My Documents“) liegen. Andernfalls werden die Bibliotheken nicht gefunden.



Unten: An den RBBB wird der USB-BUB angeschlossen, dieser via USB an den Computer.



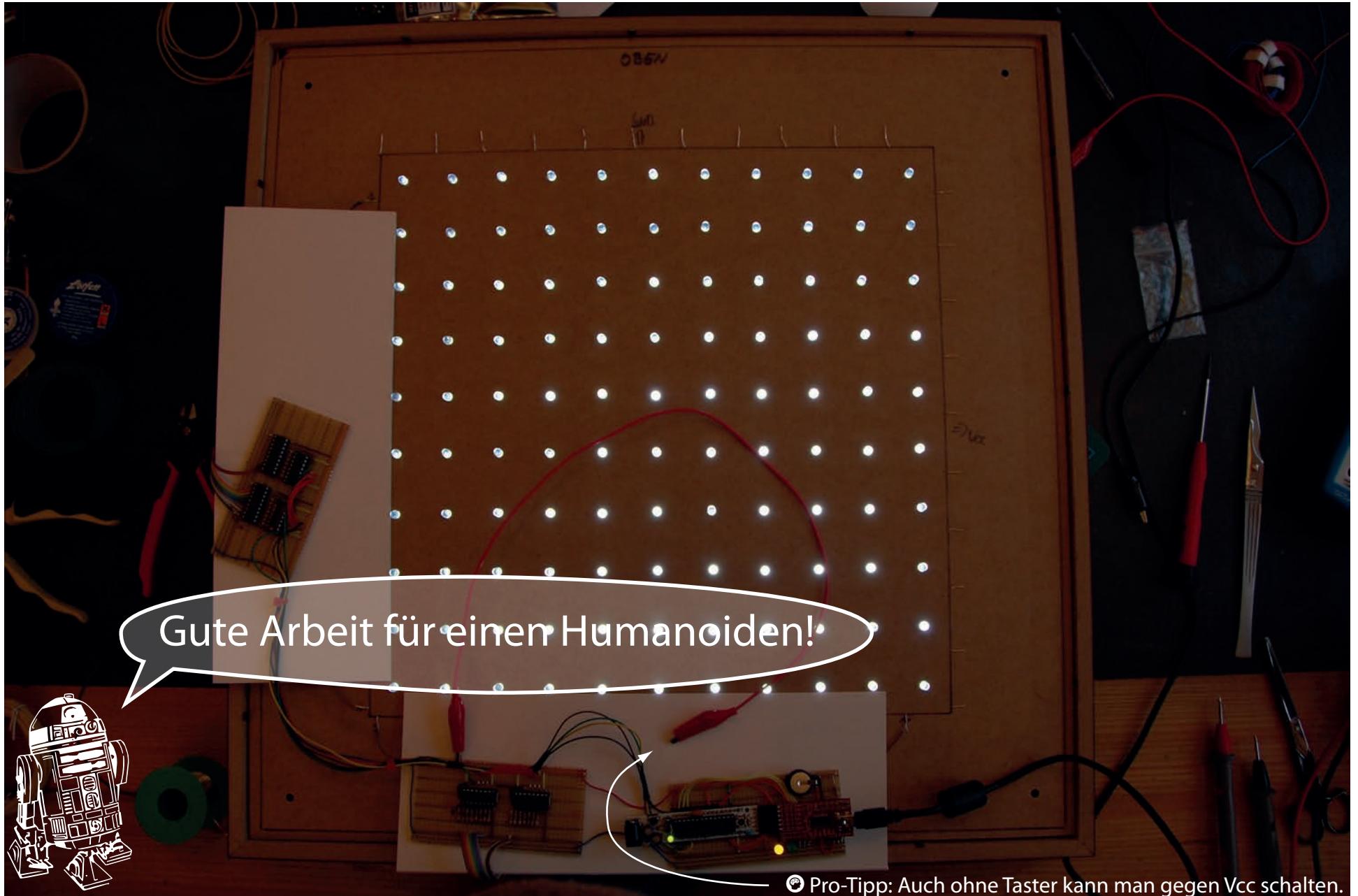
① Board auswählen: Arduino Duemilanove UNO w/ ATmega328

② Serial Port auswählen: irgendwas mit USB (je Rechner verschieden).

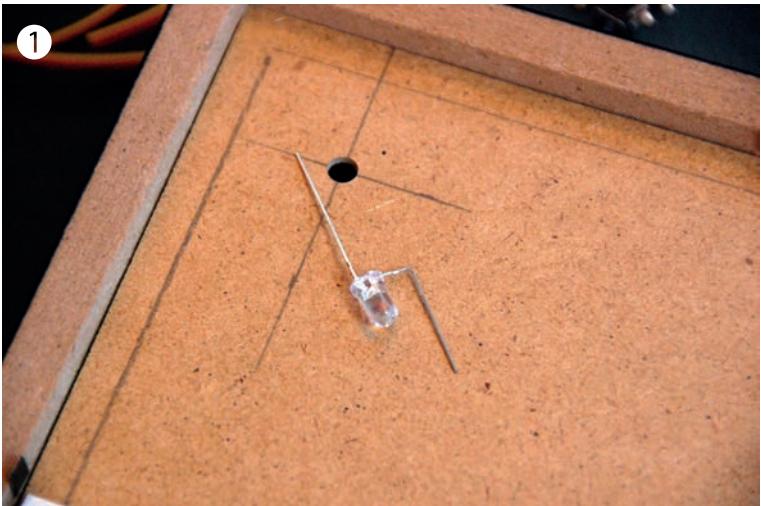
③ Qlockthree.pde (.ino) öffnen

④ Auf das RBBB hochladen

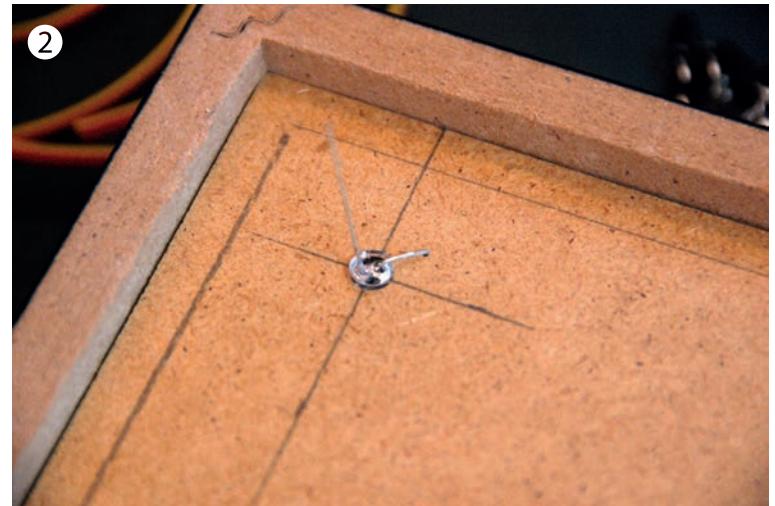
Es werde Licht!



Eck-LEDs anschliessen

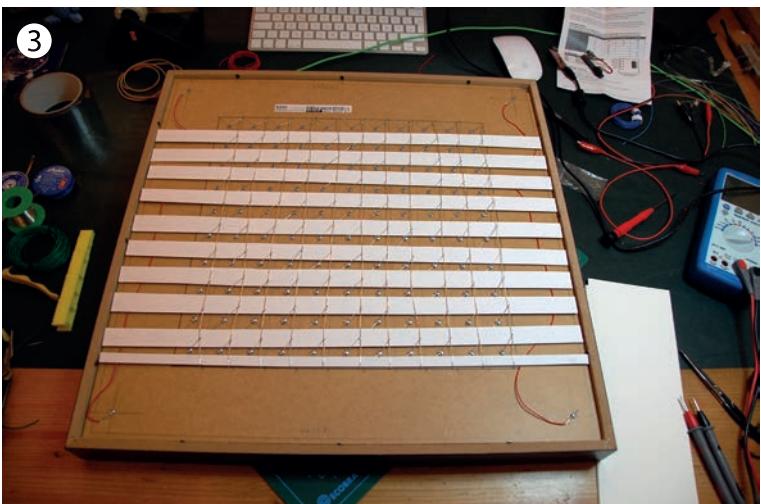
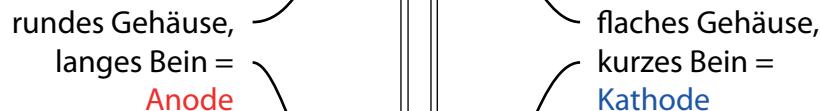


Links: Die Kathode der Eck-LEDs wird so umgebogen, daß sie durch das kleine Loch passt.

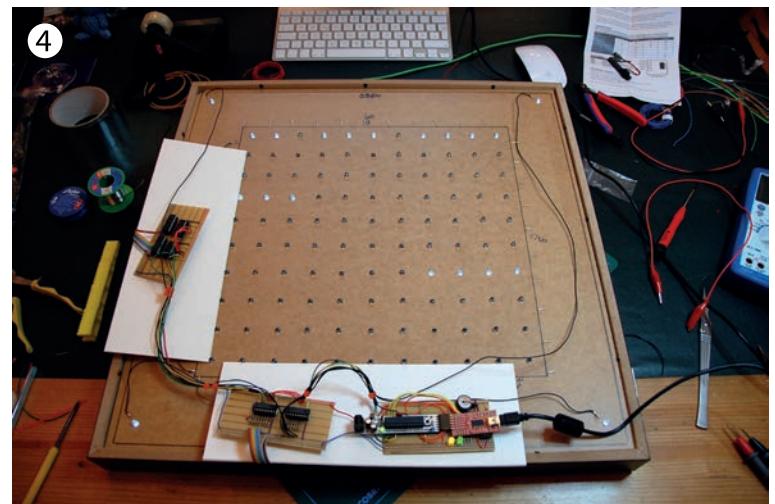


Oben: Die LEDs werden in die Löcher gesteckt.

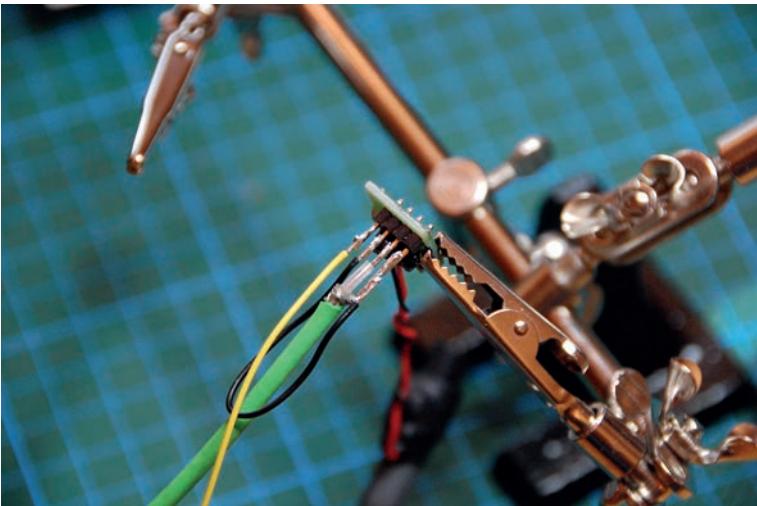
Unten: Die Anoden der LEDs werden an die Matrix angelötet. Oben links an die 1. Reihe, oben rechts an die 2., unten rechts an die 3. und unten links an die 4.



Rechts: Die Kathoden werden an den Kathoden-Multiplexer angelötet. (Reihenfolge: Oben links, oben rechts, unten rechts, unten links.)

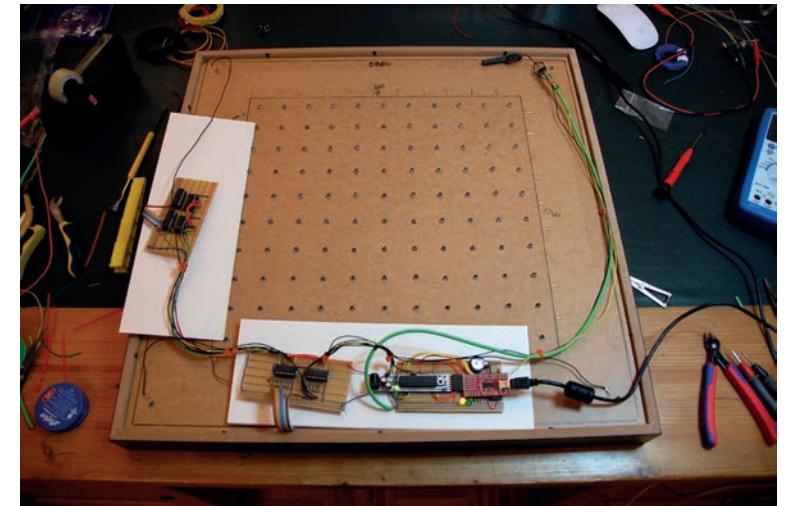


DCF77-Empfänger anschliessen



Oben: Dem DCF77-Empfänger von Pollin liegt ein falsches Datenblatt bei. Bei neueren Bestellung liegt ein zweites Datenblatt außerhalb der antistatischen Verpackung bei - das ist dann richtig, aber immer noch schwer zu interpretieren.

Oben zu sehen: Links ist die gelbe Stromversorgung (die im neuen Datenblatt auf 3 Volt abgesenkt wurde). Der zweite Pin von rechts ist die Signalleitung. Der Empfänger reagiert empfindlich auf Störungen aller Art, besonders durch das hochfrequente Multiplexing. Daher wird er räumlich weit weg montiert (oben rechts) und die Signalleitung abgeschirmt. Statt einem abgeschirmtem Kabel kann man auch ein normales Kabel nehmen und in Alufolie einwickeln, die dann auf Ground gelegt wird. Die zwei verbleibenden Pins gehen auf Ground, einer zur Stromversorgung und der andere ist „PON“, womit man den Empfänger schlafen legen kann (was wir aber nicht benutzen).

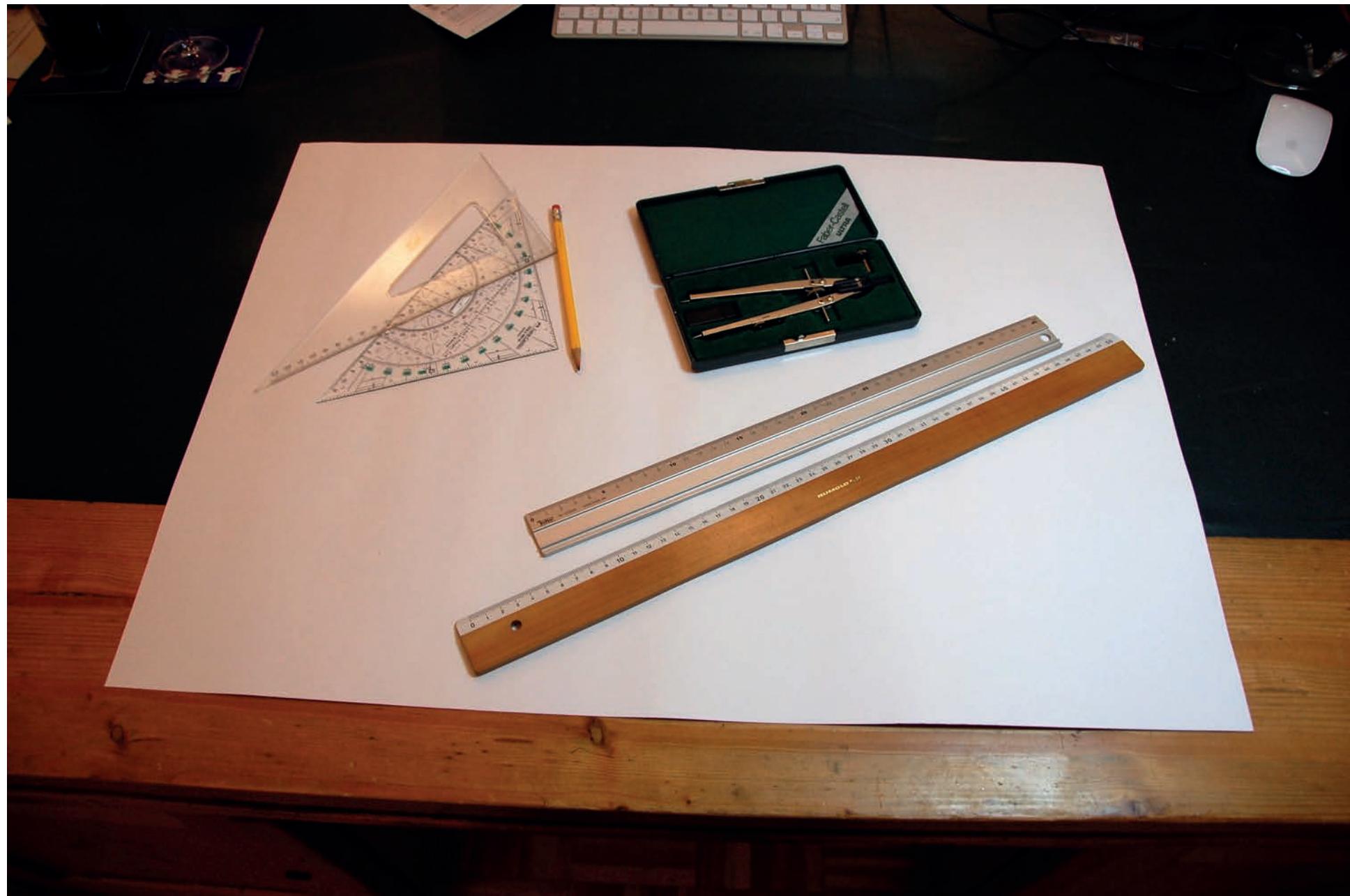


Oben: Der DCF77-Empfänger wird an die Steuerplatine angeschlossen.

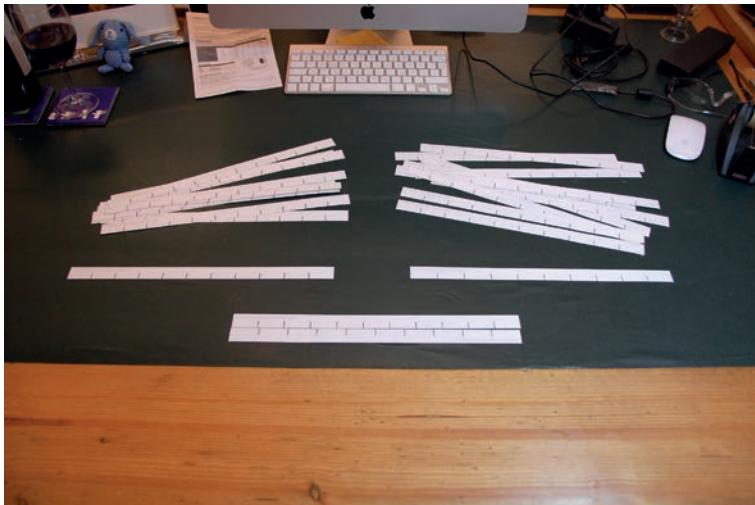
⌚ Pro-Tipp: Die Qualität des empfangenen Signals lässt sich gut an der LED ablesen. Da das Zeitsignal mit einem Impuls von 100 ms Länge (für eine 0) und 200 ms Länge (für eine 1) pro Sekunde übertragen wird, muß die LED im Gleichtakt mit der LED von der Real-Time-Clock leuchten. Einmal pro Minute muß sie für eine Sekunde dunkel bleiben, das ist der Synchronisations-Zeitpunkt. Wenn die LED wild flackert ist der Empfang gestört (durch den Computermonitor oder Ähnliches).

Mit der Elektronik sind wir jetzt (fast) fertig! Noch fehlen die Taster, aber die heben wir uns bis zum Schluß auf. Da die Taster ja mit dem Gehäuse und der Steuerplatine verbunden sind, ist das Öffnen/Schließen des Ribba-Rahmens dann mühsamer. Also machen wir erst alles andere fertig, und zum Schluß kommen dann die Taster. Va bene?

Lichtgitter bauen

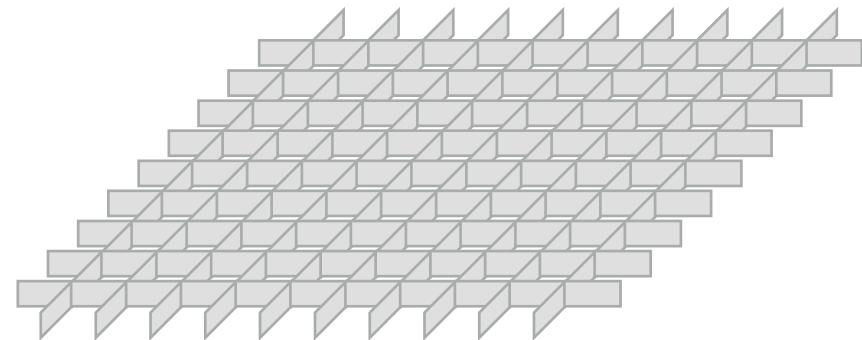


Lichtgitter bauen (fortgesetzt)

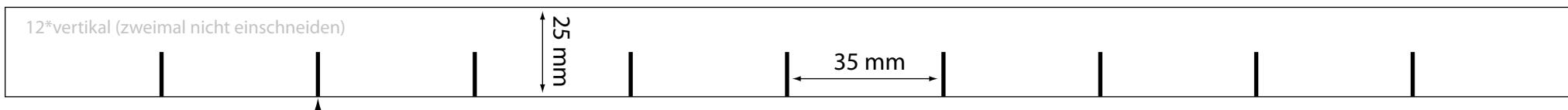
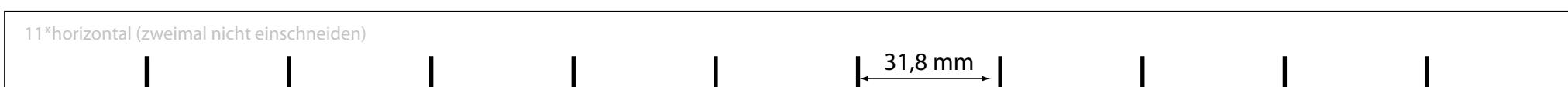


Das Lichtgitter soll verhindern, daß die LEDs auf benachbarte Buchstaben leuchten. Dazu schneiden wir aus dem Fotokarton (hier damit man etwas sieht in weiß, aber besser ist schwarz) 2,5 cm hohe Streifen mit einer Länge von 35 cm. Diese Streifen unterteilen wir in 11 Teile (auf 11 Streifen) und in 10 Teile (auf 12 Streifen). Entlang diesen Unterteilungen schneiden wir die Streifen bis zur Hälfte zweimal ein. Dann stecken wir die Streifen geschickt ineinander und erhalten das unten abgebildete Gitter.

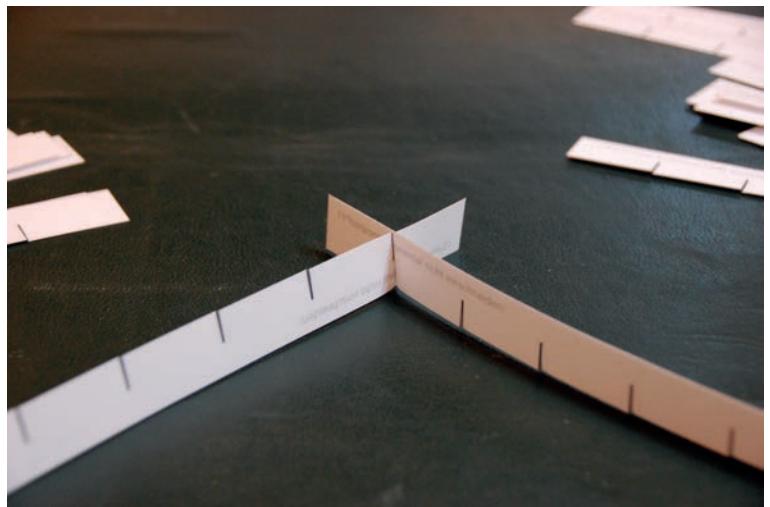
Das ist einer der fummeligsten Arbeiten überhaupt, also ruhig bleiben!



350 mm

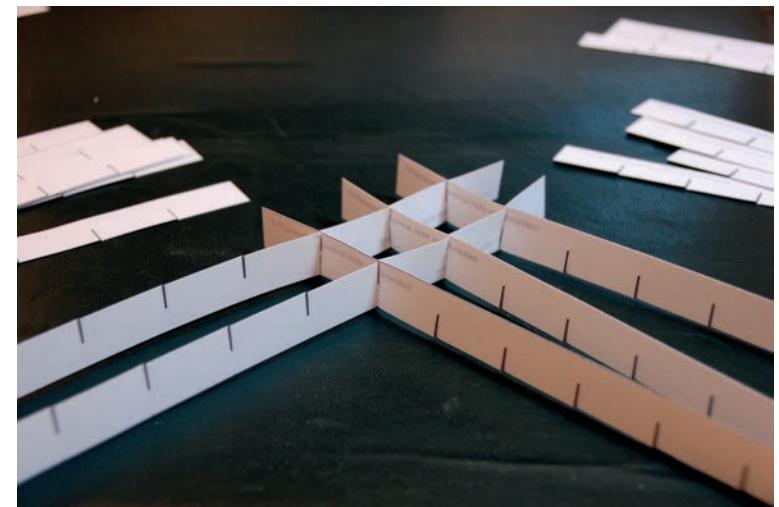


Lichtgitter bauen (fortgesetzt)

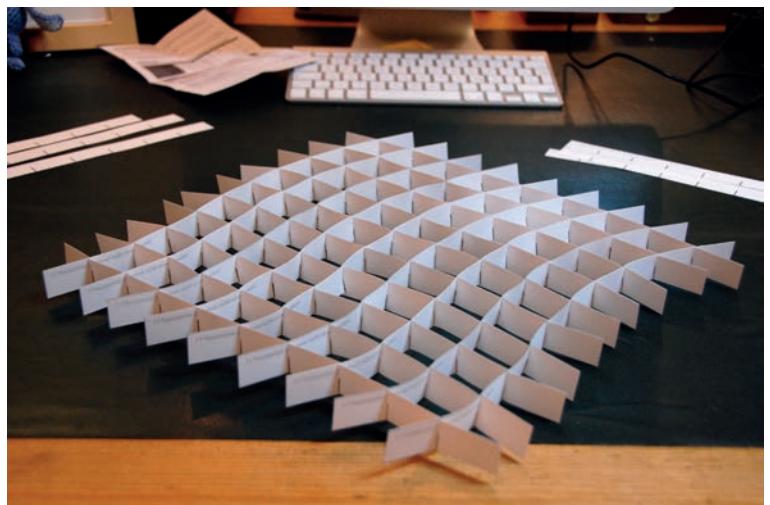


Links und rechts: Die zur Hälfte eingeschnittenen Gitterstreifen werden ineinander gesteckt.

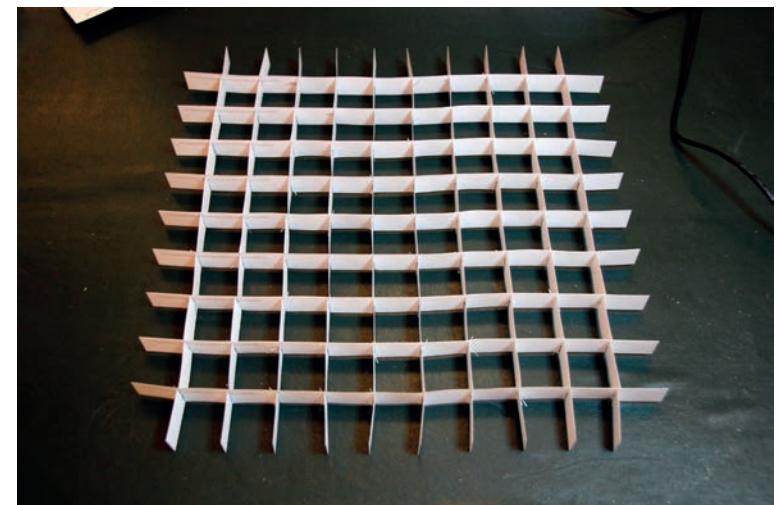
⌚ Pro-Tipp: Wenn man die Möglichkeit hat, die Gitterstreifen zu Laminieren: unbedingt nutzen! Sie werden dadurch wesentlich stabiler und reißen nicht so schnell aus.



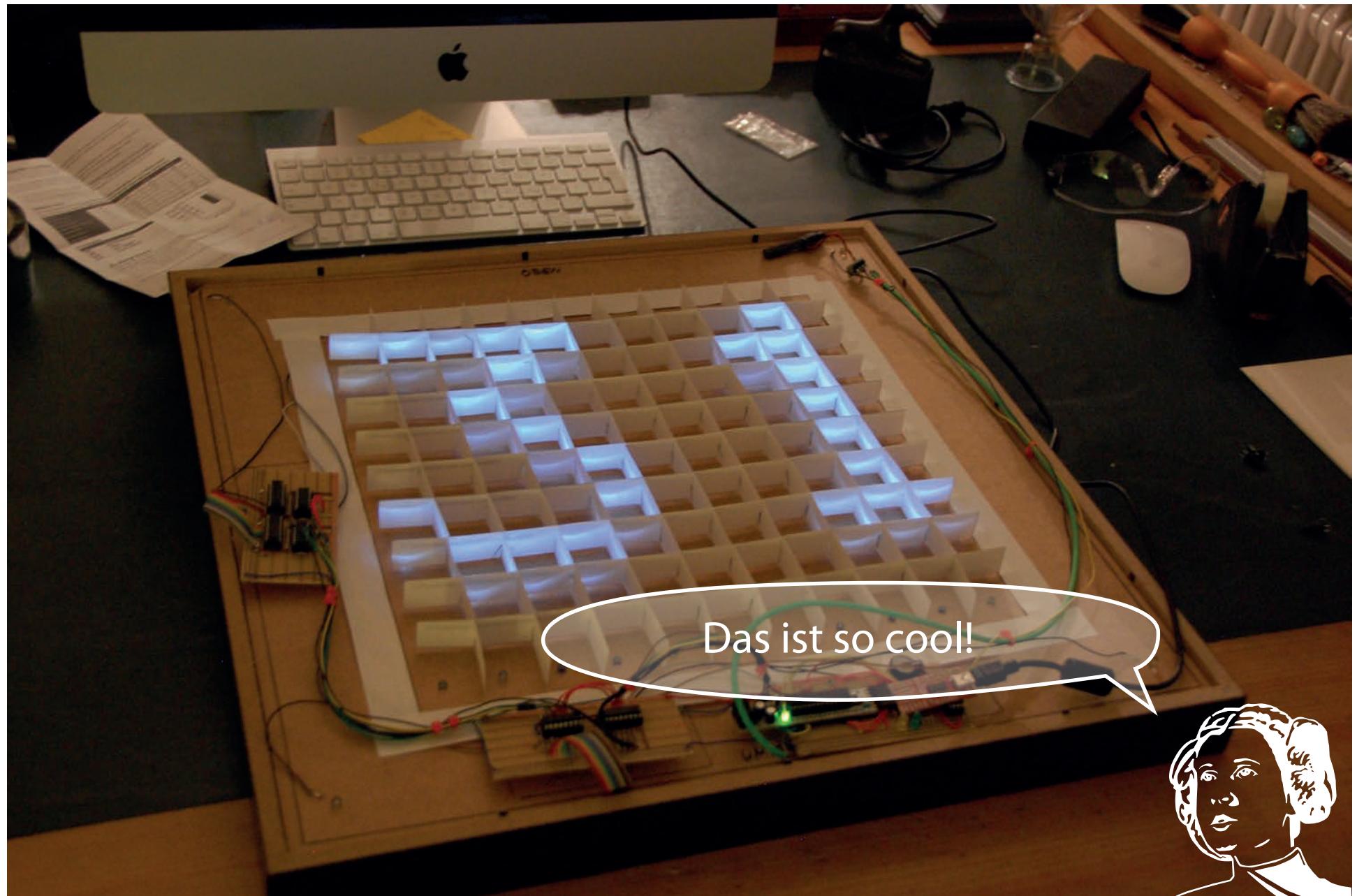
Unten: So sieht ein Gitter aus, wenn man den Schnitt nur einmal ausführt! Das ganze Gitter spannt, da das Papier zwar dünn ist, aber eben doch etwas Platz im Schnitt braucht.



Unten: So sieht es besser aus. Durch den Doppelschnitt haben die Streifen Platz und das Gitter ist schön regelmäßig. Zur Belohnung kann man es mal aufsetzen (nächste Seite).



Lichtgitter bauen (fortgesetzt)

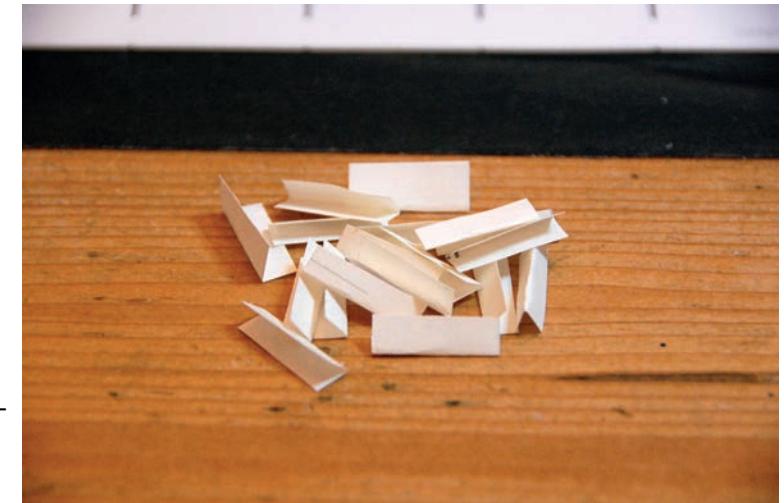


Lichtgitter bauen (fortgesetzt)

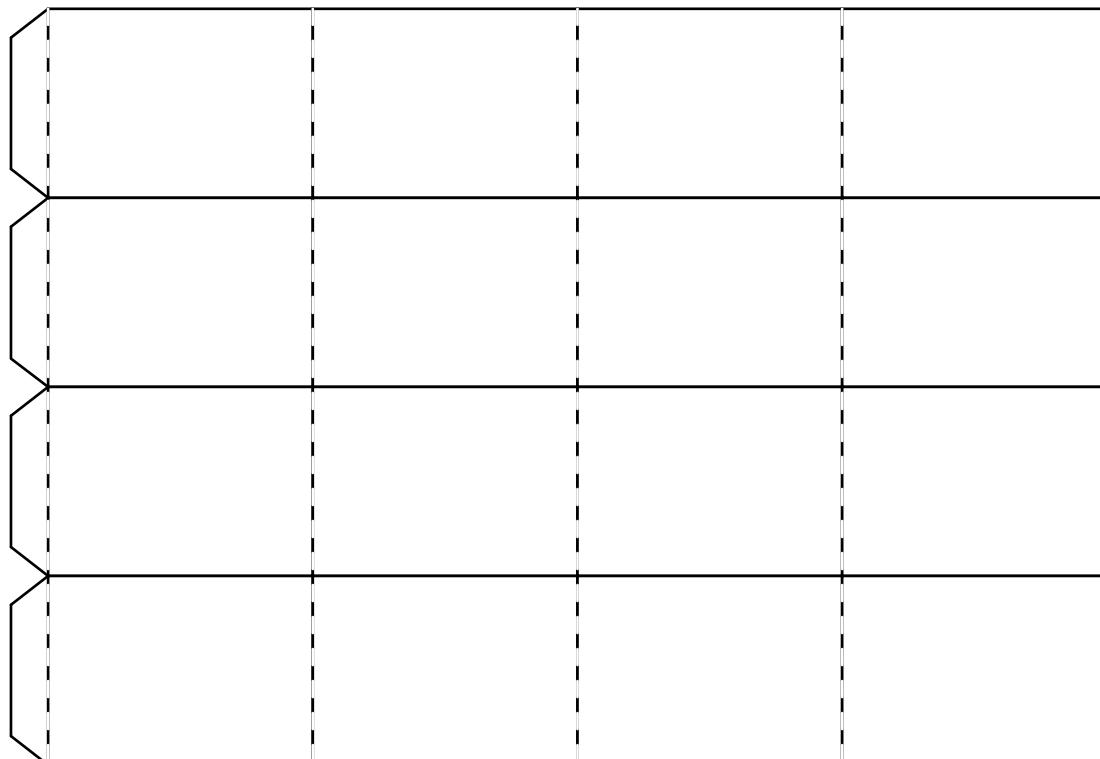


Links: Jetzt müssen noch die „Außenwände“ montiert werden. Dazu brauchen wir ein paar 2,5 cm hohe Papierstreifen und Klebstoff (und etwas Geduld).

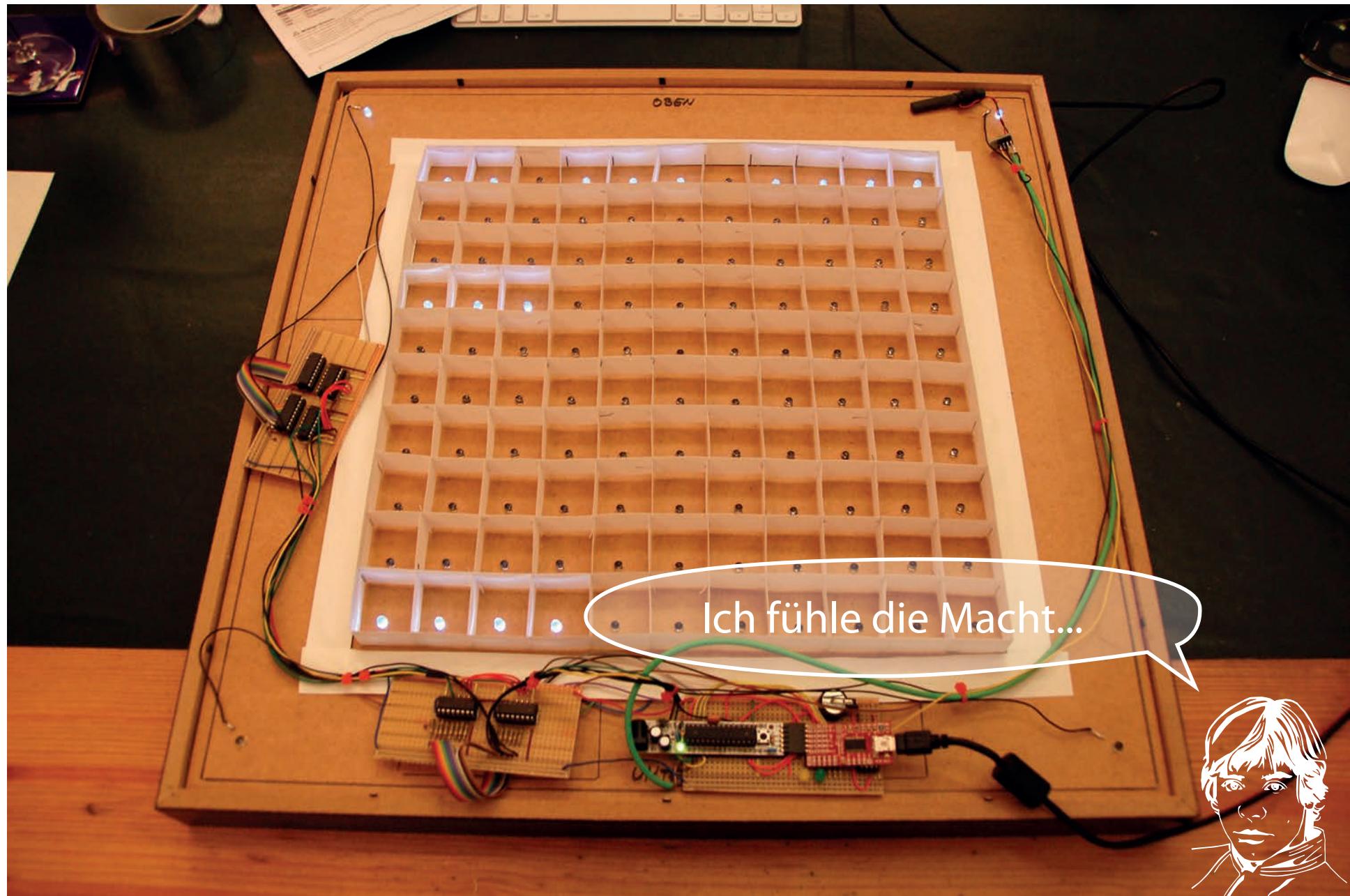
Rechts: Aus den Papierstreifen machen wir uns einen Haufen 2,5 cm hohe „Ecken“, mit denen wir dann die Streifen für die Außenwände an die frei stehenden Gitterstreifen kleben.



Für die Eck-LEDs brauchen wir vier separate Kästchen. Das Muster rechts ist im Maßstab 1:1. Einfach ausschneiden, an den gestrichelten Linien falten und zusammenkleben.



Lichtgitter bauen (fortgesetzt)



Folie kaschieren



Folie kaschieren (fortgesetzt)

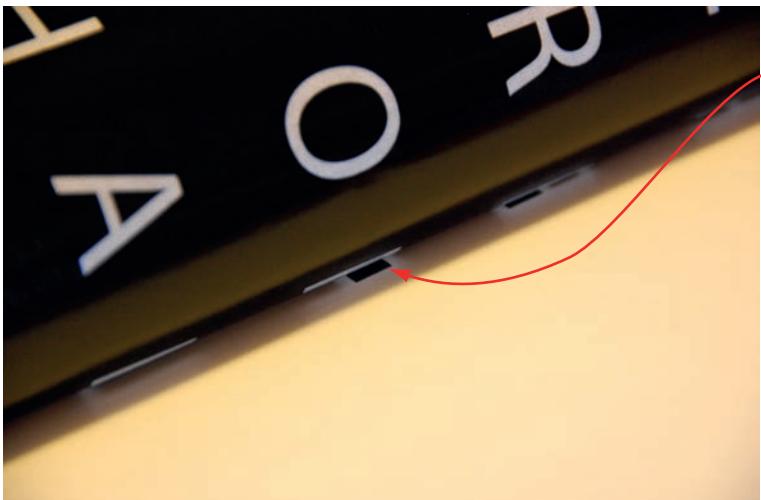


Zuerst wird die Glasplatte mit dem Glasreiniger sauber geputzt. Was später zwischen Folie und Glas ist, kann nie wieder entfernt werden.

Dann wird die Glasplatte mit Spülwasser eingeschmoddert. Tipp: Einfachstes Spülmittel ohne Zusätze wie Handbalsam etc. nehmen.



Hier ist Sauberkeit das Allerwichtigste! Jeder Fussel, jedes Haar wird nachher zwischen Folie und Glas stören! Alles sauber entstauben und putzen (Glas, Eimer, Schwamm, Haare; lange Haare zum Pferdeschwanz binden etc.) Wenn möglich, neuen Schwamm oder frisches Handtuch verwenden.



Bei dem Abziehen der Folie von dem Trägermaterial aufpassen, es bleiben gerne die feinen Innenteile von Buchstaben hängen.

Am Besten zieht man die Trägerfolie über eine Tischkante nach unten weg. Eine zweite Person ist hier hilfreich!



Folie kaschieren (fortgesetzt)

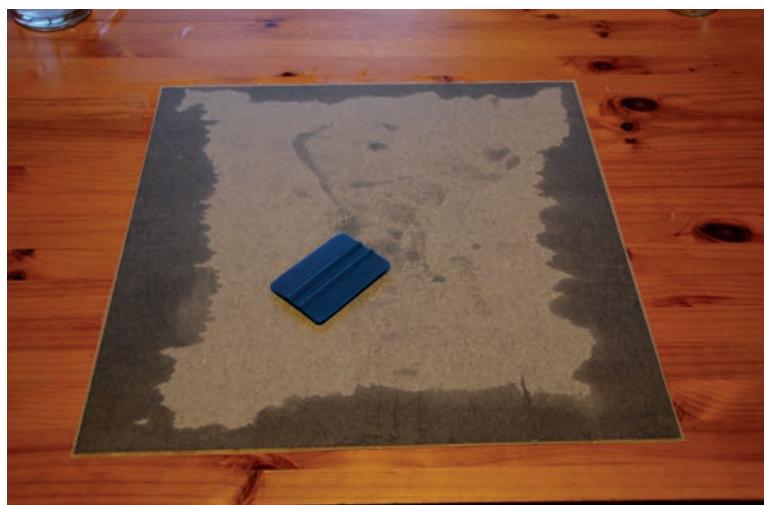


Links: Die abgezogene Folie.

Rechts: Die Folie wird auf das mit Spülmittel eingeschmoldderte Glas gelegt. Durch das Spülmittel klebt sie noch nicht fest, sondern ist positionierbar. Wer (so wie ich) viel Spülmittel genommen hat, muß eventuell erst einmal auf der Vorderseite etwas Spülmittel herausdrücken.



Unten: Auf der Rückseite drücken wir jetzt von der Mitte ausgehend mit dem Rakel das Spülmittel zwischen Folie und Glas heraus (min. 10 Minuten, nicht aufgeben).



Unten: Die fertige Frontplatte mit der aufkaschierten Folie liegt zur Probe auf dem Lichtgitter. Das sieht doch schon ganz gut aus, oder?

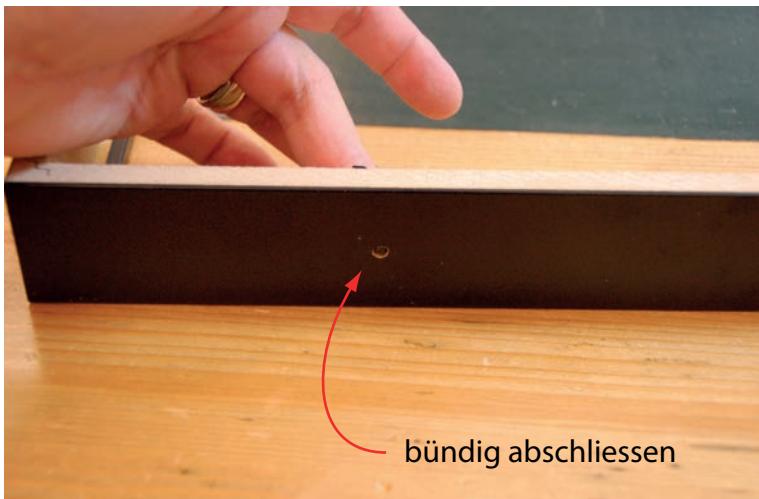


Finale: Taster montieren



In den Ribba-Rahmen bohren wir drei Löcher. Die Taster sollten nicht alle drei nebeneinander liegen, da man dann später nicht mehr weiss, welcher Taster welche Bedeutung hat. Besser ist es, den Modus-Taster auf die eine und die Stell-Taster auf die andere Seite zu legen.

Unten: Die Rahmenverdünnung sollte so sein, daß der Taster bündig mit dem Rahmen abschließt. Keine Bange, man kann ihn dann immer noch drücken.



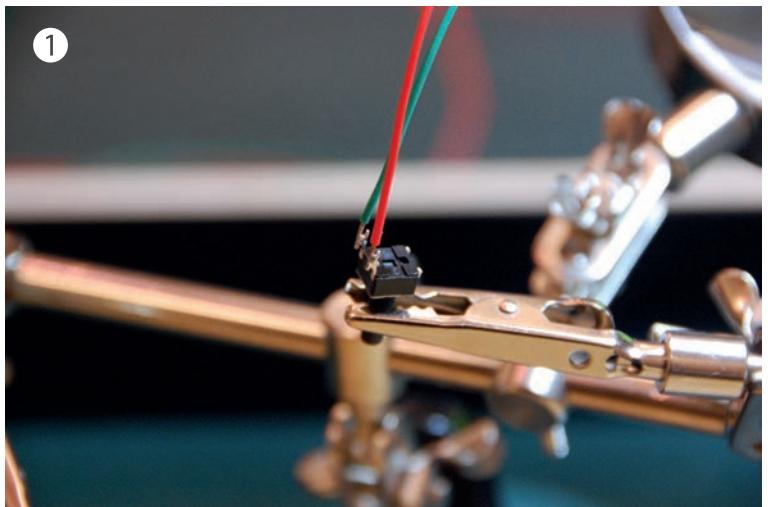
Rechts: Achtung! Die Taster haben eine schmale und eine breite Seite! Sie schalten zwischen der kurzen Seite, die breiten Seiten bilden einen Kurzschluß.



Oben: Die Taster sind eine winziges Fitzelchen zu kurz, wir müssen daher um das Bohrloch herum die Rahmenstärke etwas dünner machen. Der Rahmen ist aus Presspappe, daher geht das mit einem Teppichmesser ganz gut.

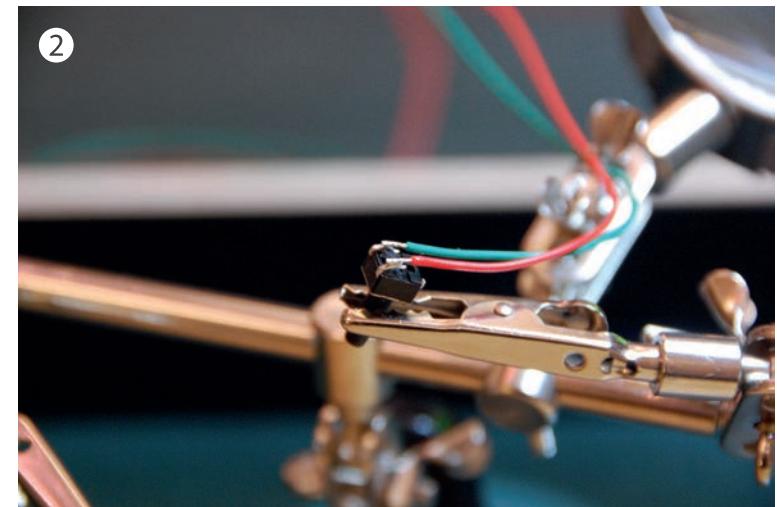


Finale: Taster montieren (fortgesetzt)



1

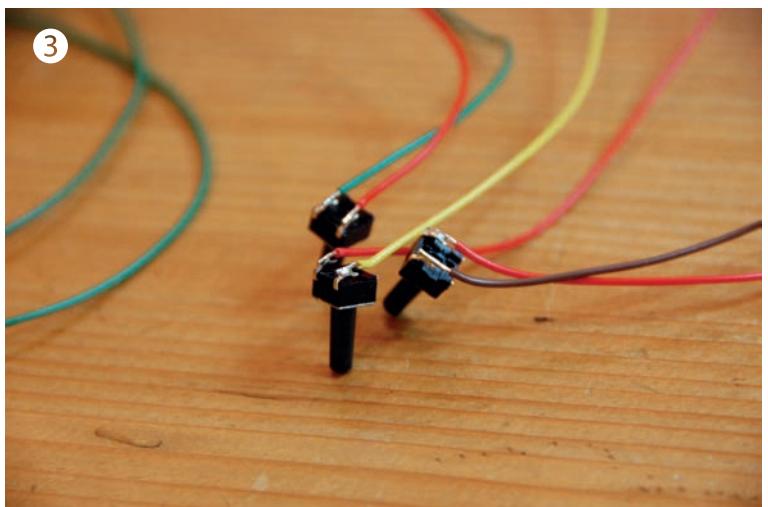
Links: Zwei Beine zwicken wir ab, an die anderen beiden Beine löten wir ein rotes Kabel (das geht später an Vcc) und eine buntes Kabel.



2

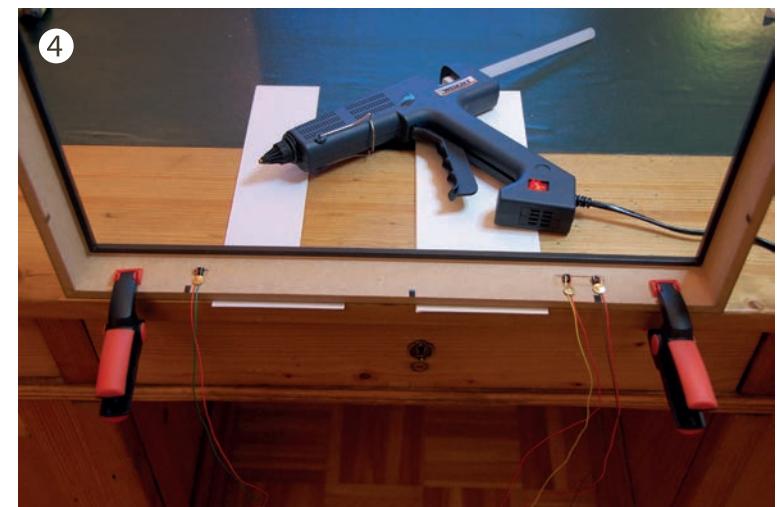
Rechts: Die verlöteten Beine biegen wir auf den Taster um. Der Taster ist später zwischen dem Innenrahmen (Abstandshalter) und der Außenwand, daher sollte er so wenig Platz wie möglich einnehmen.

Unten: Unsere drei Taster, alle mit dem roten Kabel für Vcc und bunten Kabeln für den Anschluß an das RBBB.



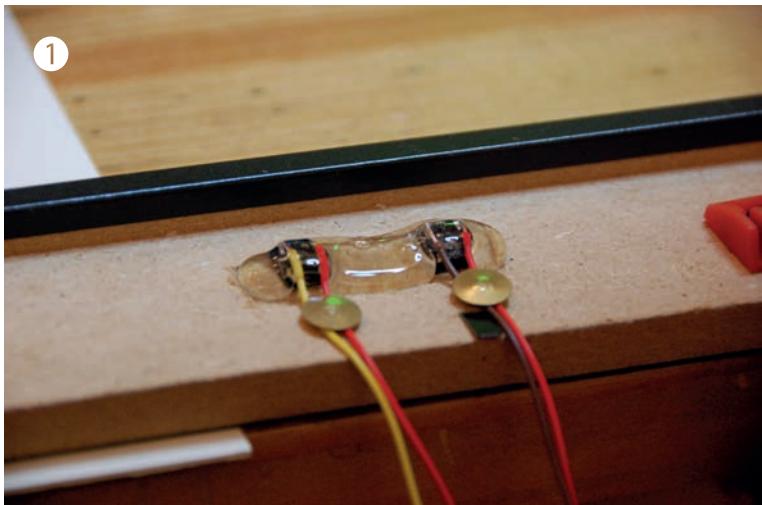
3

Rechts: Um die Taster mit Heißkleber einzugießen wird der Rahmen fest eingespannt, die Taster kann man gut mit Reißzwecken fixieren. Reste des Passepartou dienen als dünne Abstandshalter, damit die Taster nicht von unten herausgedrückt werden.



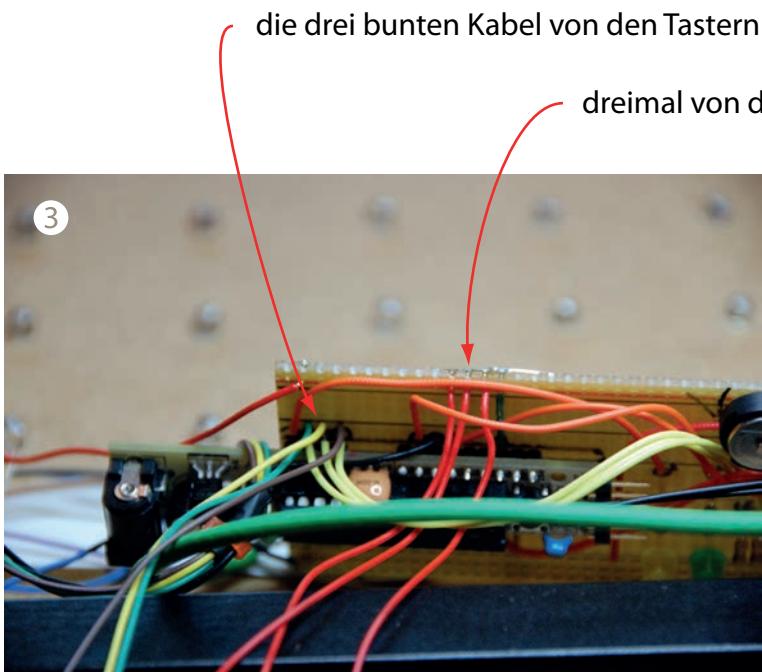
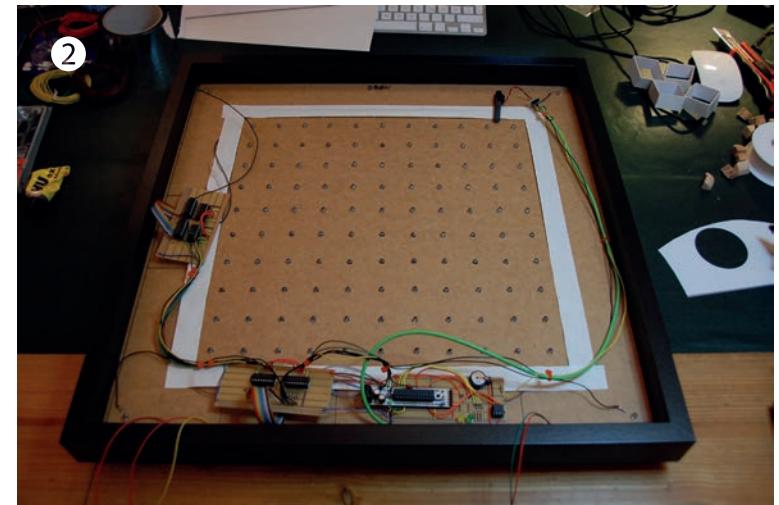
4

Finale: Taster montieren (fortgesetzt)

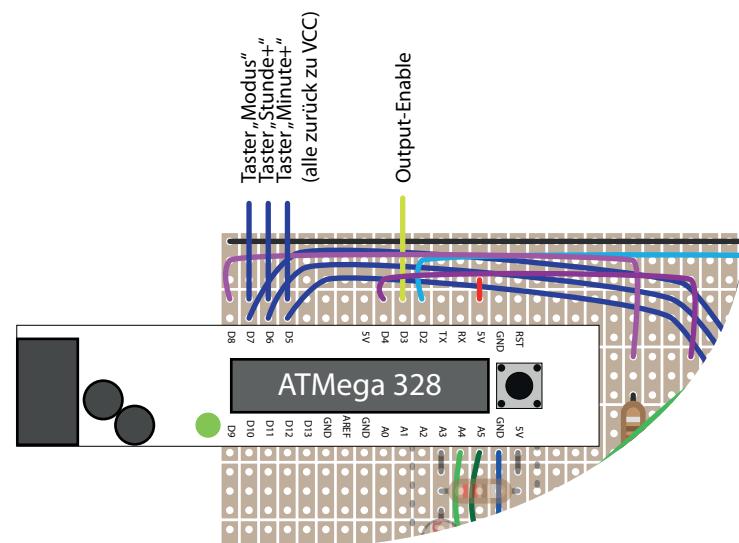


Links: Die Taster werden jetzt mit ordentlich Heißkleber „eingegossen“. Sie brauchen dann eine ganze Weile zum Abkühlen, also geduldig sein.

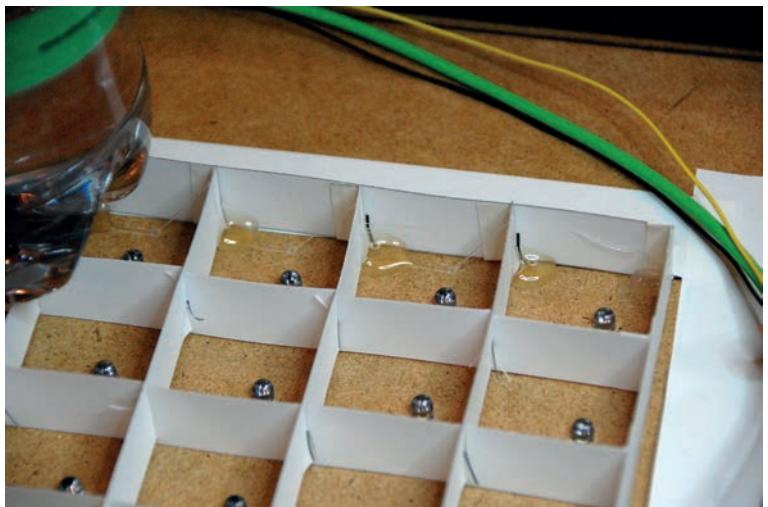
Rechts: Für die „Endmontage“ legen wir die Rückwand in den Rahmen ohne Frontplatte, wie sie später auch liegen wird. Sie ist mit Büchern unterlegt, damit sie ein wenig nach oben gedrückt wird.



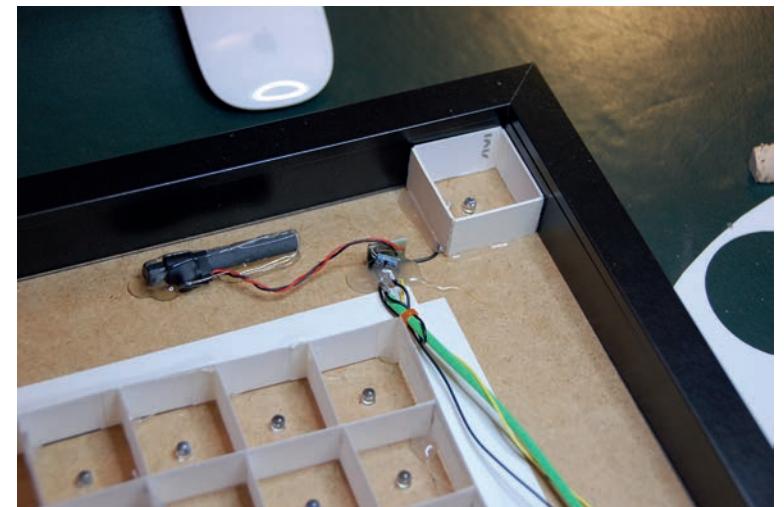
Links: Jetzt löten wir die Kabel der Taster an die Steuerplatine. Rote Kabel gehen irgendwo an Vcc, die bunten an das RBBB.



Finale: Lichtgitter befestigen und Füße schneiden



Links: Das Lichtgitter wird mit Heißkleber gepunktet. Einmal rundherum.



Rechts: Die Ecken werden genauso festgeklebt. Dabei kann man auch gleich den DCF77-Empfänger und seine Antenne festkleben. Die Antenne sollte dabei parallel zum Rahmen ausgerichtet sein.

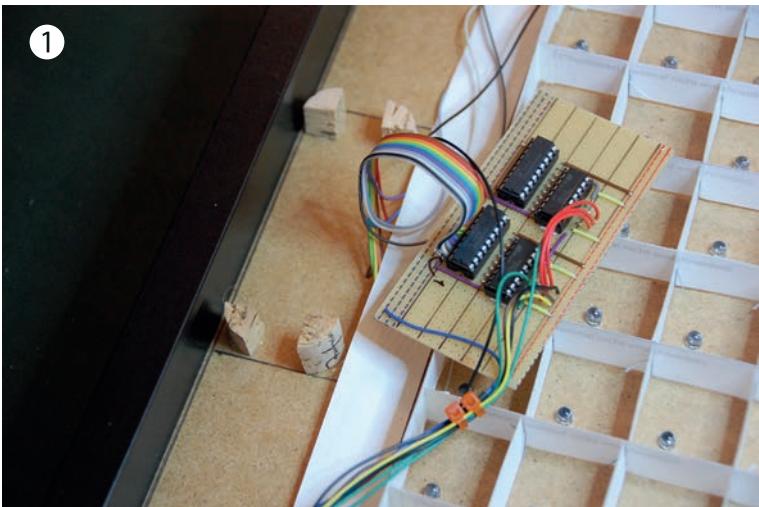
Unten: Als Abstandshalter für die Platinen bereiten wir kleine Füße aus Kork vor. Am Besten ist hierzu der Korken eines Barolos aus der Langhe geeignet.



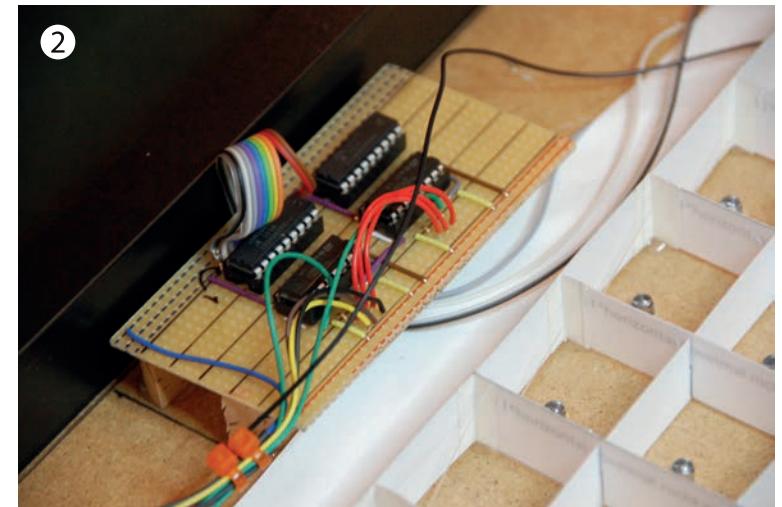
Rechts: Den Korken zerschneiden wir in drei Scheiben, die drei Scheiben in jeweils vier „Kuchenstücke“.



Finale: Platinen festkleben

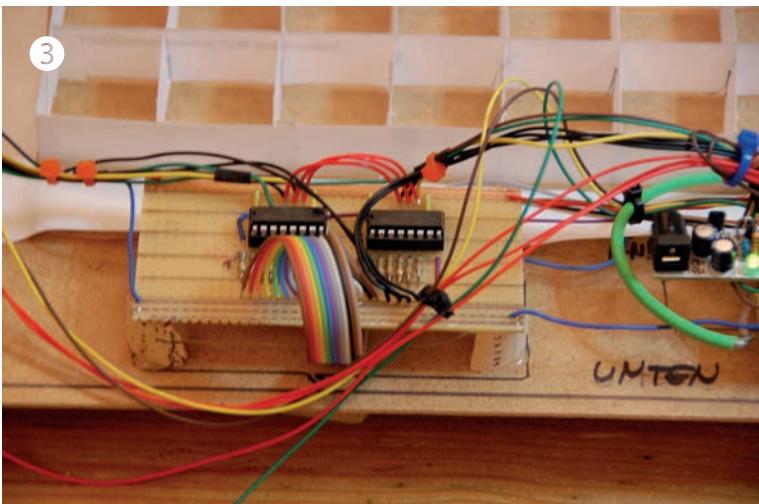


Links: Vier von den Beinen werden auf die Rückwand geklebt.

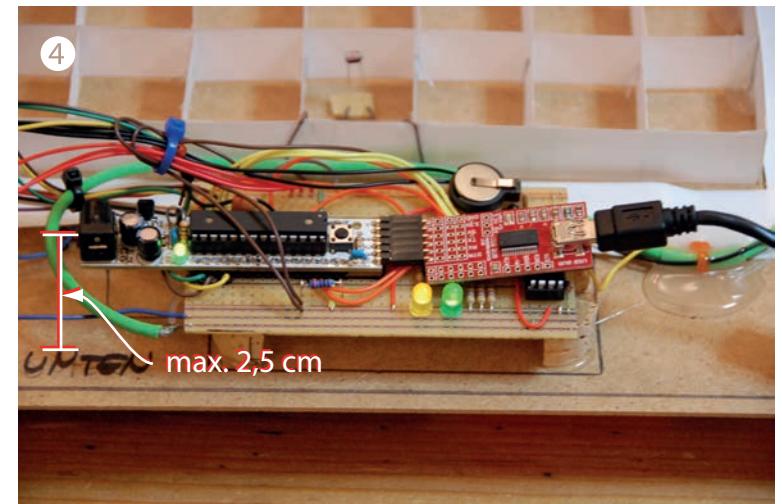


Oben: Die Platine wird auf die Korkbeine geklebt.

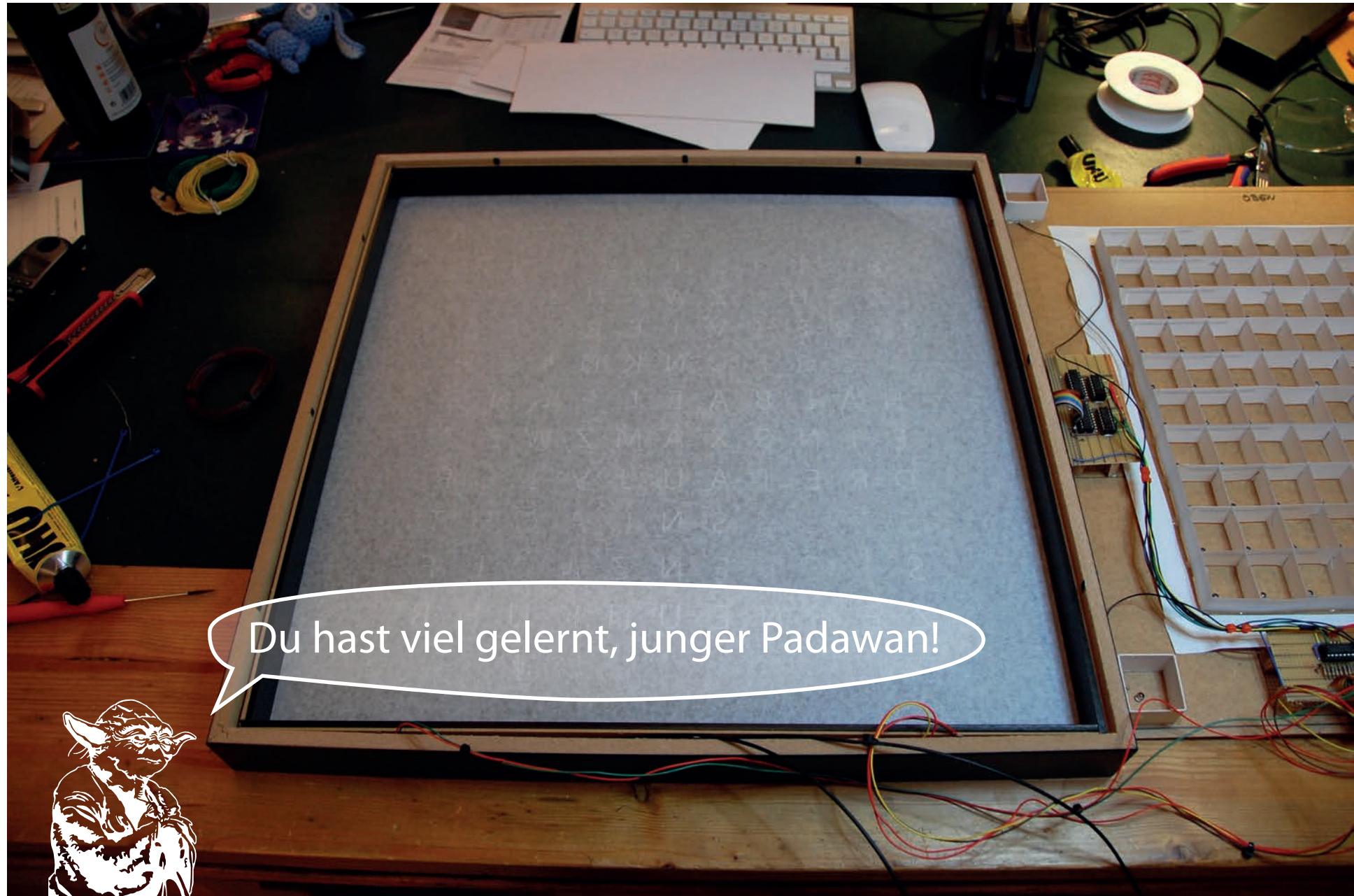
Unten: Analog zum Anoden-Multiplexer wird auch der Kathoden-Multiplexer verklebt.



Rechts: Die Steuerplatine wird aufgeklebt. Achtung! Der höchste Teil der Platine (der Power-Jack) darf an der Oberkante maximal 2,5 cm von der Bodenplatte entfernt sein.



Finale: Frontplatte einsetzen

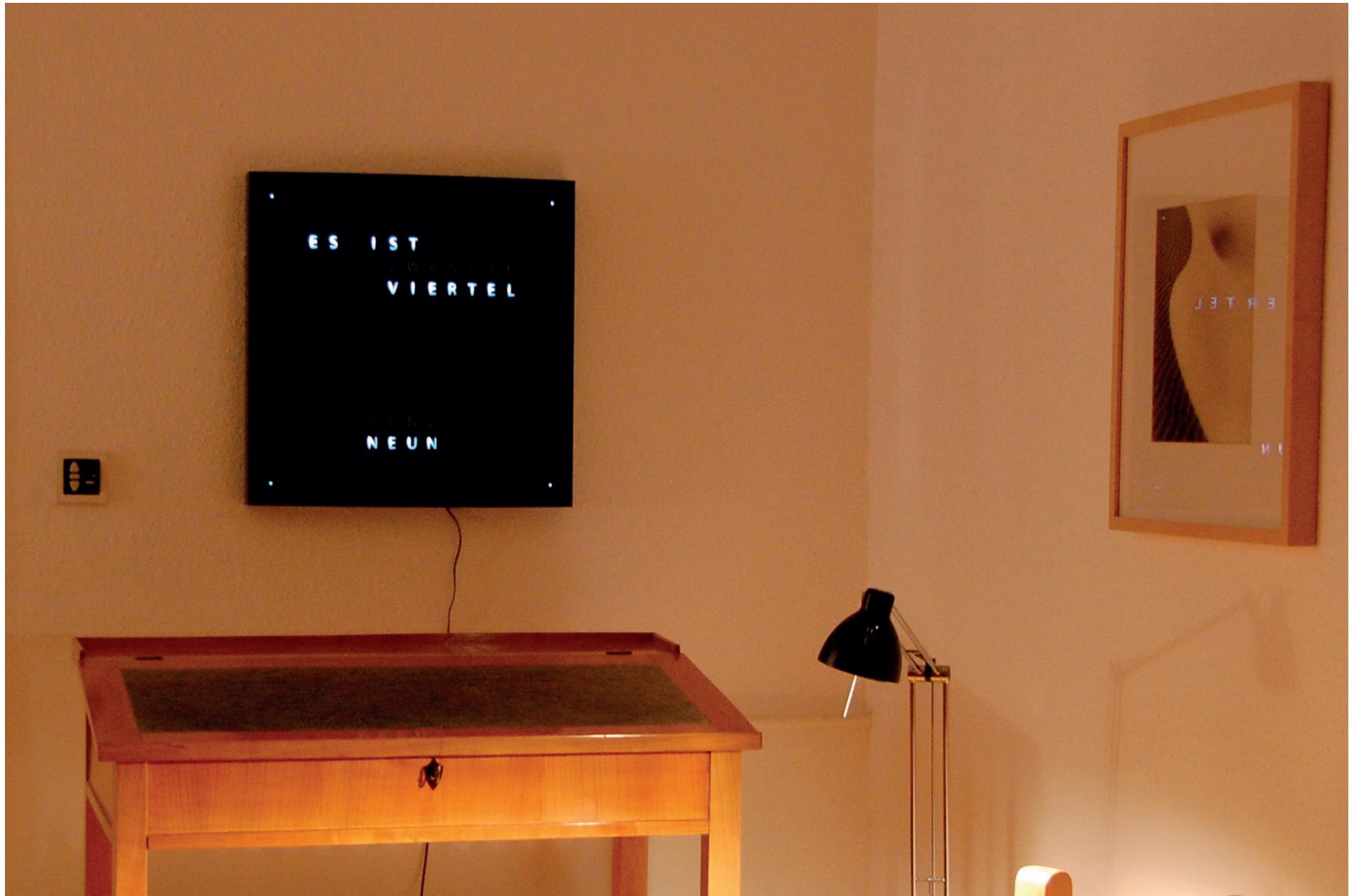




The digital clock displays the time in German words. The digits are represented by the following words:
E S K I S T A F Ü N F
Z E H N Z W A N Z I G
D R E I V I E R T E L
V O R F U N K N A C H
H A L B A E L F Ü N F
E I N S X Ä M Z W E I
D R E I A U J V I E R
S E C H S N L A C H T
S I E B E N Z W Ö L F
Z E H N E U N K U H R

GRATULATION!

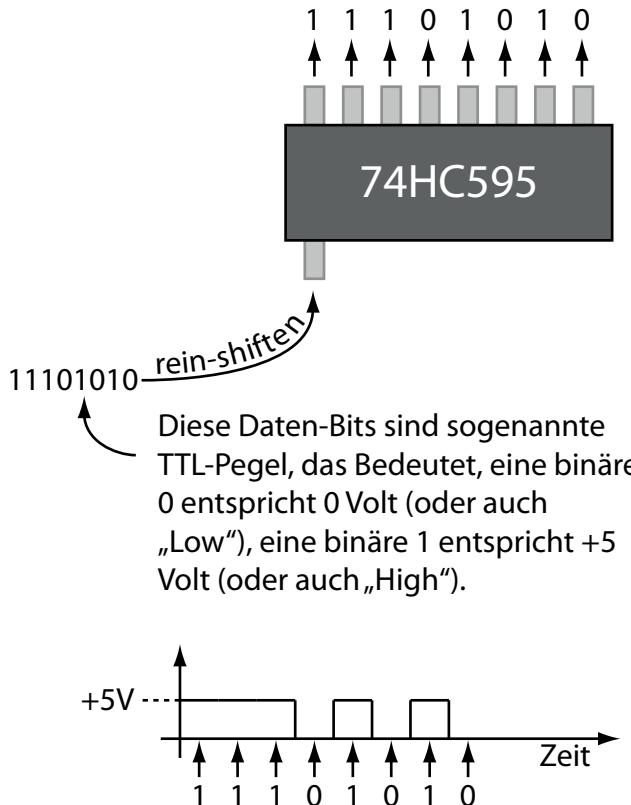
Appendix A - wie funktioniert die QLOCKTWO?



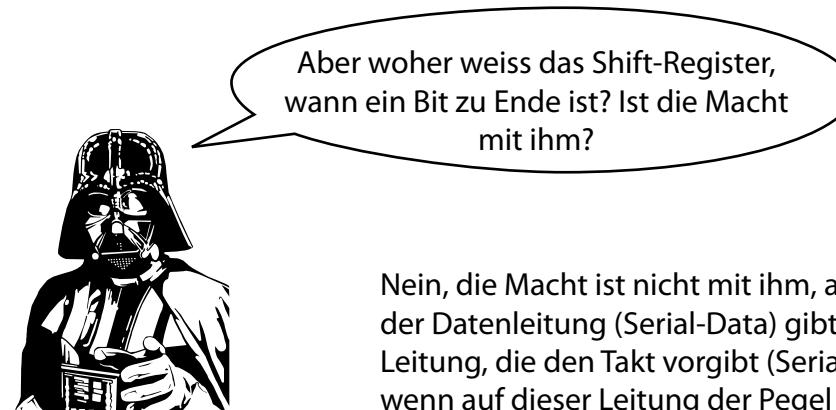
LED-Matrix-Multiplexing: das Shift-Register

Die Matrix der QLOCKTWO besteht (neben den vier Eck-LEDs) aus einem Gitter von LEDs mit 10 Reihen und 11 Spalten. Das sind insgesamt 114 LEDs. Es gibt aber keinen Microcontroller mit 114 Ausgängen um jede LED individuell ansteuern zu können. Der von uns verwendete ATmega328-Microcontroller von Atmel hat z. B. 14 digitale und 6 analoge Ein- und Ausgänge. Um die Anzahl der Ausgänge zu erhöhen, bedient man sich häufig und gerne dem 74HC595-Shift-Register.

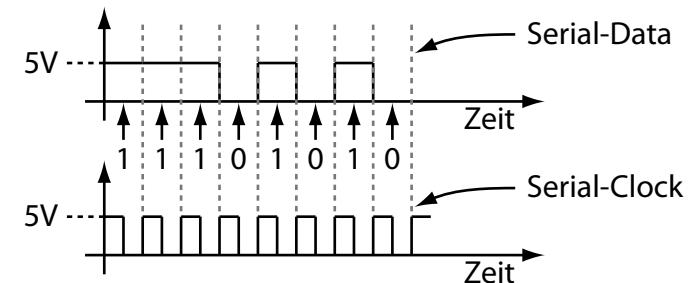
Bei einem Shift-Register handelt es sich um einen Baustein, in den Daten (Bits) über eine Leitung hineingeschoben werden (geshiftet), und der diese Daten dann an vielen Ausgängen bereitstellt. Dadurch haben wir plötzlich 8 Ausgänge mehr!



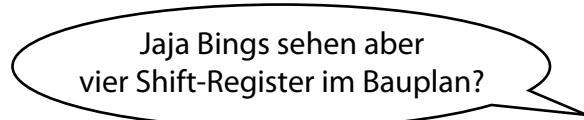
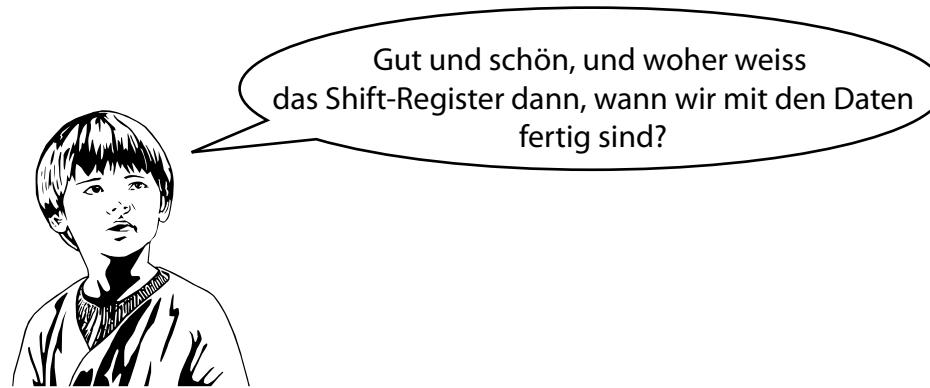
Diese Daten-Bits sind sogenannte TTL-Pegel, das bedeutet, eine binäre 0 entspricht 0 Volt (oder auch „Low“), eine binäre 1 entspricht +5 Volt (oder auch „High“).



Nein, die Macht ist nicht mit ihm, aber zusätzlich zu der Datenleitung (Serial-Data) gibt es noch eine Leitung, die den Takt vorgibt (Serial-Clock). Immer, wenn auf dieser Leitung der Pegel von Low auf High springt, ist ein Bit zu Ende.



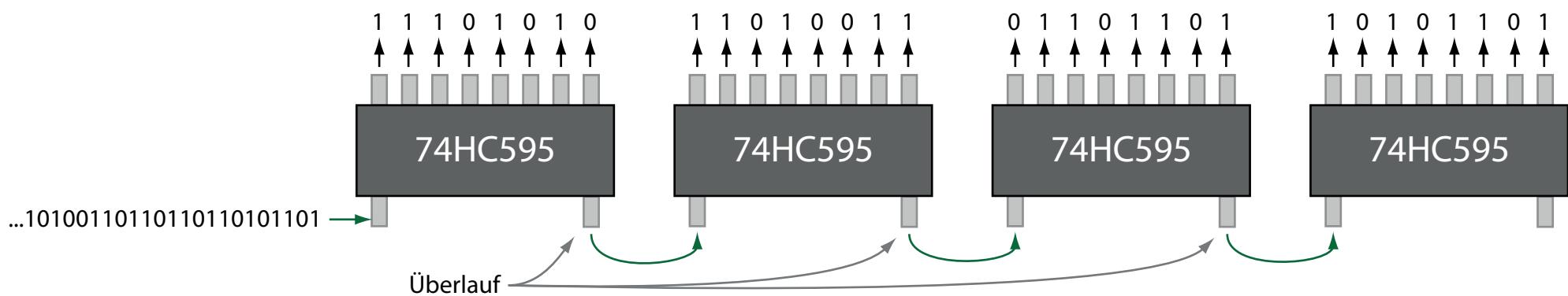
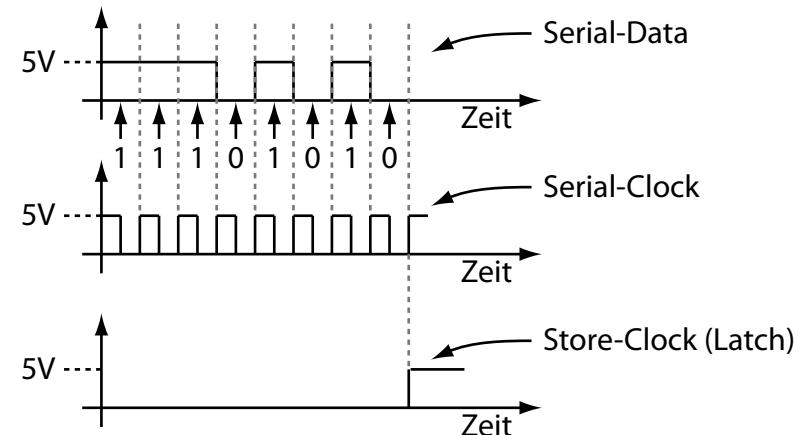
LED-Matrix-Multiplexing: das Shift-Register (fortgesetzt)



Wir brauchen mehr als 8 Ausgänge. Zum Glück reicht aber eine Daten-Leitung. Jedes Shift-Register hat einen „Überlauf“. Schiebt man mehr Daten hinein, als ein Shift-Register aufnehmen kann, werden die überschüssigen Daten aus dem Überlauf herausgeschoben. Diese kann man dann in das nächste Shift-Register reinschieben.



Zusätzlich zu der Datenleitung (Serial-Data) und der Takt-Leitung (Serial-Clock) gibt es noch einen Riegel (Latch oder Store-Clock). Immer, wenn diese Riegel-Leitung „low“ ist, dürfen Daten geschrieben werden. Ist sie „high“, werden die Daten auf den Ausgängen ausgegeben.

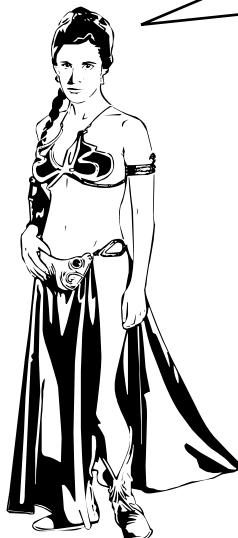
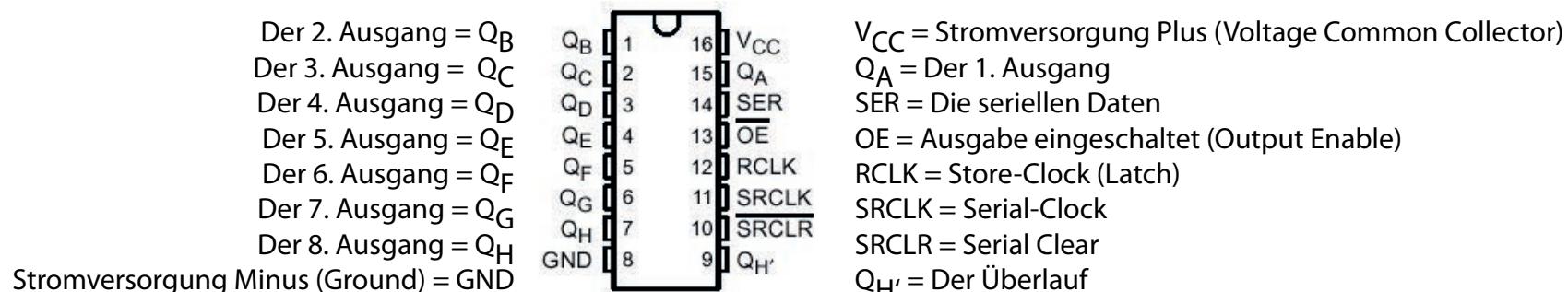


LED-Matrix-Multiplexing: das Shift-Register (fortgesetzt)

So wie unten wird das Shift-Register im Datenblatt dargestellt. Diejenigen, die ganz genau aufgepasst haben, werden zwei Pins entdecken, die bislang noch nicht beschrieben worden sind.

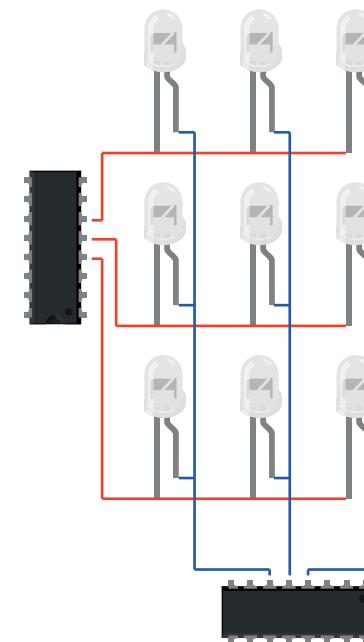
OE = Output Enable (Ausgabe eingeschaltet) ist ein Pin, mit dem man alle Ausgänge ein- und ausschalten kann. Der Strich über OE bedeutet das OE ein invertierter Pin ist. Das wiederum bedeutet, OE ist aktiv wenn der Pegel auf Low ist.

SRCLR = Serial Clear (Daten löschen) ist ein Pin, mit dem alle Inhalte des Registers löschen kann. Auch SRCLR ist ein invertierter Pin.



Toll: Drei Seiten Erklärungen,
und wir haben für 114 LEDs gerade mal
 $4 \cdot 8 = 32$ Ausgänge!

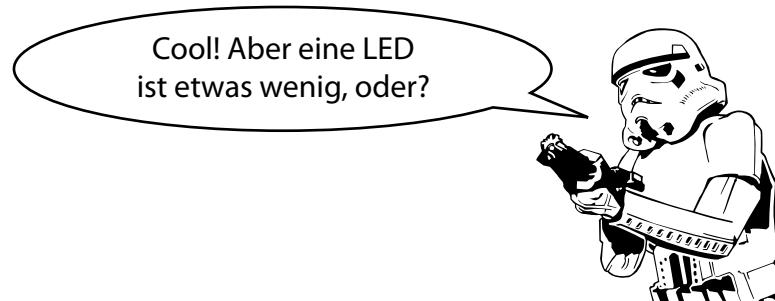
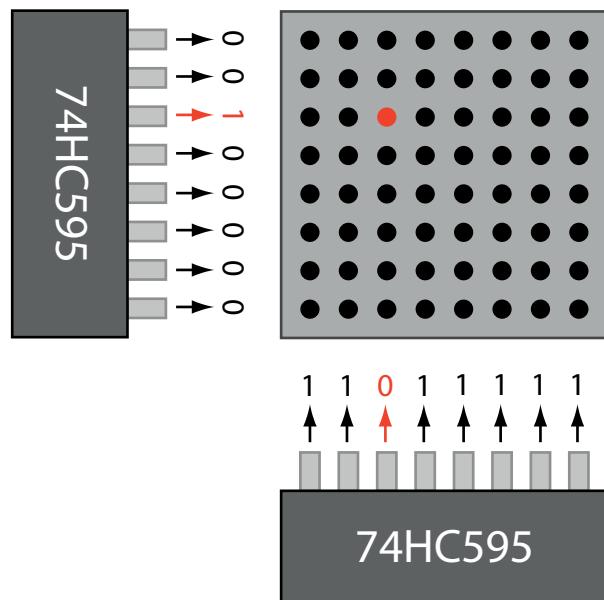
Das stimmt, und daher benutzen wir
einen Trick: Nicht jede LED bekommt
ihren eigenen Ausgang, sondern eine
ganze Reihe von LEDs bekommen einen
Ausgang. Und eine Spalte auch! (Rechts
schematisch am Beispiel mit 3 mal 3
LEDs.)



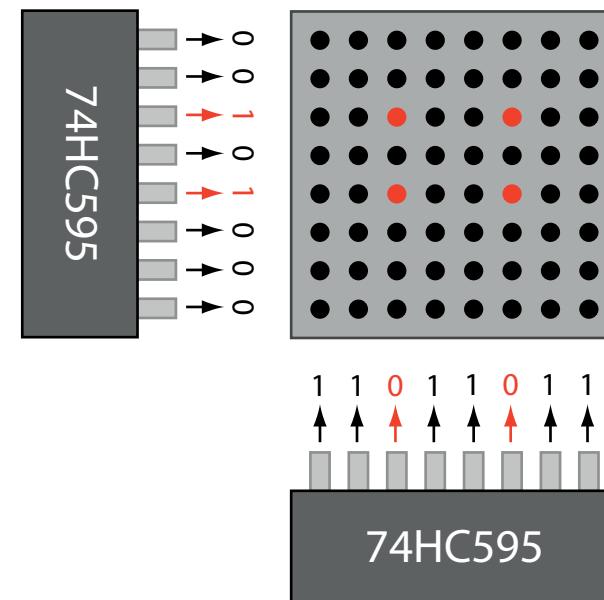
LED-Matrix-Multiplexing: die Matrix

Der Einfachheit halber betrachten wir im Folgenden nur noch eine 8x8-Matrix, aber das Prinzip gilt auch für unsere (theoretisch bis zu) 16x16-Matrix.

Wir können jetzt eine LED einschalten, indem wir auf der Seite mit den Plus-Polen der LEDs (Anoden) eine 1 ausgeben und auf der Seite mit dem Minus-Polen (Kathode) eine 0. Da eine LED eine Diode ist und den Strom nur in einer Richtung durchlässt, bleiben alle anderen LEDs dunkel, obwohl Spannung anliegt.

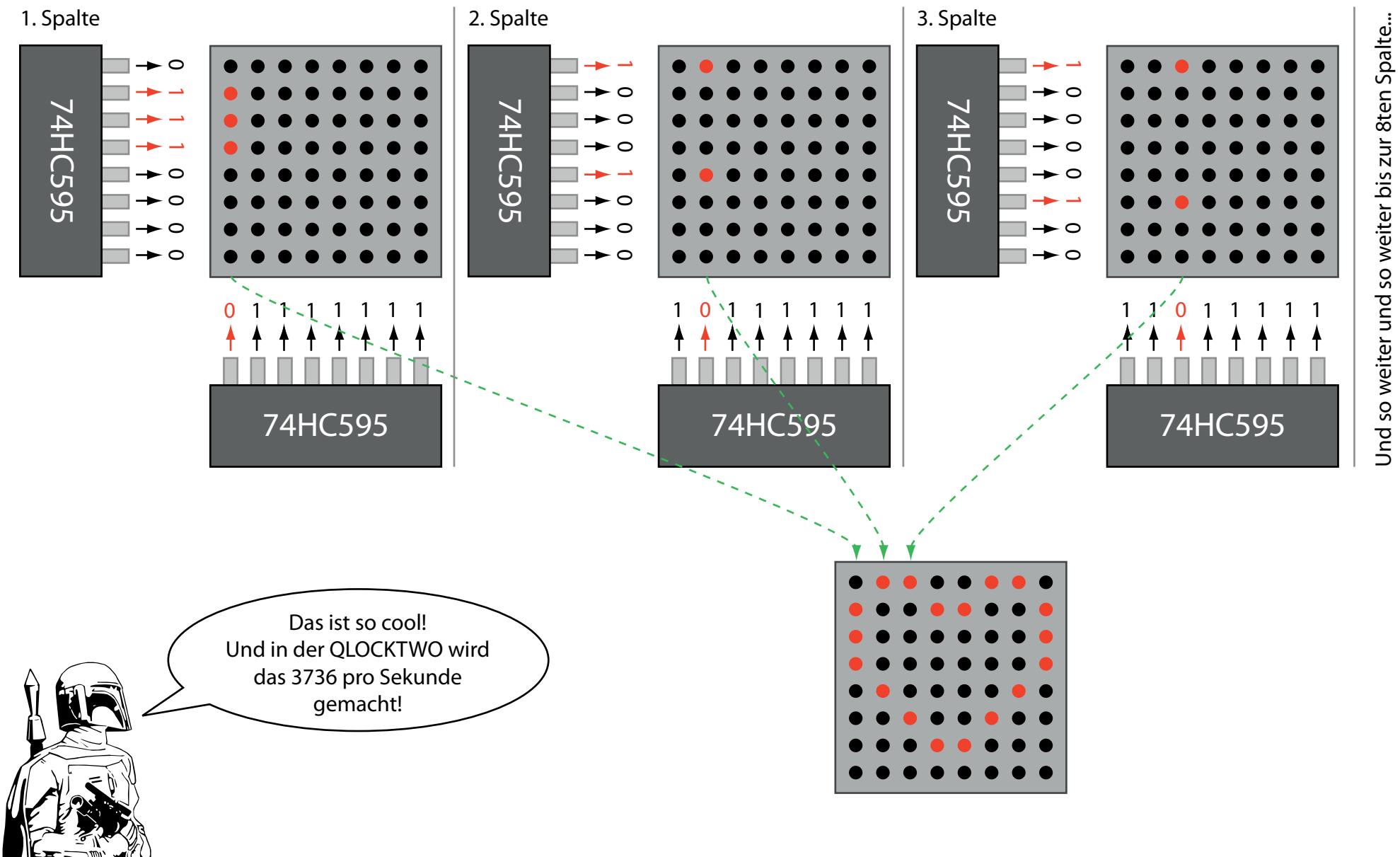


Das Problem dabei ist, schalten wir mehr als eine Reihe und Spalte ein, leuchten immer alle betroffenen LEDs! Wir könnten also nie nur die LED oben rechts und unten links leuchten lassen!



LED-Matrix-Multiplexing: die Matrix (fortgesetzt)

Damit wir dennoch beliebige Muster ausgeben können, schalten wir immer nur eine Spalte ein! Erst die Erste, dann die Zweite und so weiter. Und das machen wir so schnell, daß es für das menschliche Auge wie ein stehendes Bild aussieht. So macht das ein Röhrenfernseher übrigens auch. Diesen Effekt nennt man „Nachbildungswirkung“ oder „Persistence of vision“.



Darstellung der Wörter

Die Wörter, die für die Darstellung der Zeit nötig sind, werden jetzt einfach als Bit-Muster definiert, die man dann über die Shift-Register ausgeben kann. Diese Definitionen sind im Quellcode nach der Definition der Zahlen hinterlegt.

```
#define FUENF          matrix[0] |= 0b00000000111100000  
#define ZEHN           matrix[1] |= 0b111100000000000000  
#define VIERTEL        matrix[2] |= 0b0000111111100000  
#define ZWANZIG         matrix[1] |= 0b0000111111100000  
#define HALB            matrix[4] |= 0b1111000000000000  
#define DREIVIERTEL    matrix[2] |= 0b1111111111100000
```

Welche Zeile?

Welche LEDs in der Zeile?

Diese fünf Bits sind übrig

(16 Bits könnten wir ansteuern, aber in der QLOCKTWO brauchen wir nur 11)

E	S	K	I	S	T	A	F	Ü	N	F
Z	E	H	N	Z	W	A	N	Z	I	G
D	R	E	I	V	I	E	R	T	E	L
V	O	R	F	U	N	K	N	A	C	H
H	A	L	B	A	E	L	F	Ü	N	F

So weit kann man das ja
verstehen, aber was soll eine 0-te Zeile sein?
Und was soll das „|“-Zeichen?

In der Programmierung zählt man im Gegensatz zum normalen Leben von Null an. Hat man 16 Zeilen, zählt man von 0 bis 15. Das sind dann auch 16 sogenannte „Zustände“.

Das „|“-Zeichen ist ein binäres „Oder“. Es bewirkt letztlich, daß eine Zeile nicht mit dem Wort überschrieben wird, sondern nur überlagert. In der unteren Zeile im Beispiel oben stehen die Wörter „HALB“ und „FÜNF“. Wenn es halb Fünf ist, würde die „FÜNF“ das Wort „HALB“ überschreiben. Durch die Überlagerung leuchten beide Wörter zusammen.



Die Software (stark vereinfacht)

```
void loop() {  
    getDateDs1307();  
    setWords(hour, minute);  
    setMinutes(minute);  
  
    if (digitalRead(minutePlusPin) == HIGH) {  
        minute++;  
    }  
  
    if (digitalRead(hourePlusPin) == HIGH) {  
        hour++;  
    }  
  
    if (digitalRead(modePin) == HIGH) {  
        mode++;  
    }  
  
    if (mode != BLANK) {  
        writeMatrix();  
    }  
  
    digitalWrite(dcf77Led, digitalRead(dcf77Signal));  
    digitalWrite(rtcSQWLed, digitalRead(rtcSQWSignal));  
  
    const char *v = dcf77.getDateTime();  
    second = 0;  
    minute = dcf77.getMinutes();  
    hour = dcf77.getHours();  
    setDateDs1307();  
}
```

Die loop()-Schleife wird unendlich durchlaufen.
Hier lesen wir die Echtzeituhr aus.
Die Stunden auf die Matrix schreiben.
Die Minuten auf die Matrix schreiben.

Wurde der Taster für die Minuten gedrückt?

Wurde der Taster für die Stunden gedrückt?

Wurde der Taster für den Modus gedrückt?

Matrix ausgeben, wenn nicht ausgeschaltet.

Status-LED DCF77-Signal.
Status-LED Real-Time-Clock.

DCF77-Empfänger auslesen.

Zeit vom DCF77-Empfänger übernehmen.
Wieder nach oben zum Schleifenanfang.

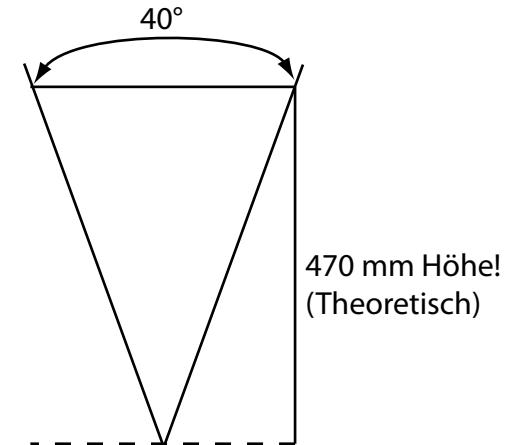
Appendix B - Ausblicke



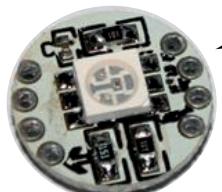
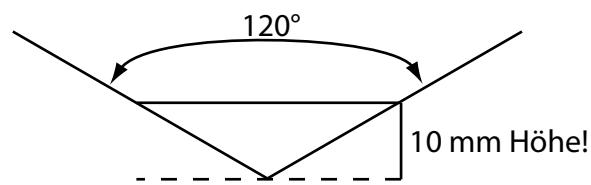
Und wie geht es jetzt weiter?
Ist das alles gewesen?

Nein! Viele Es gibt viele Möglichkeiten für die individuelle Gestaltung der eigenen Uhr. Speziell mit anderen LEDs und deren Öffnungswinkel...

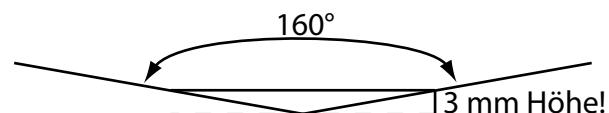
Unsere LEDs



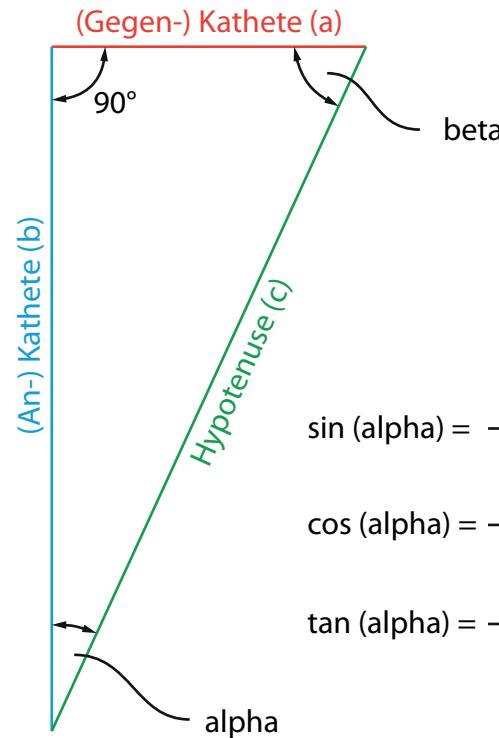
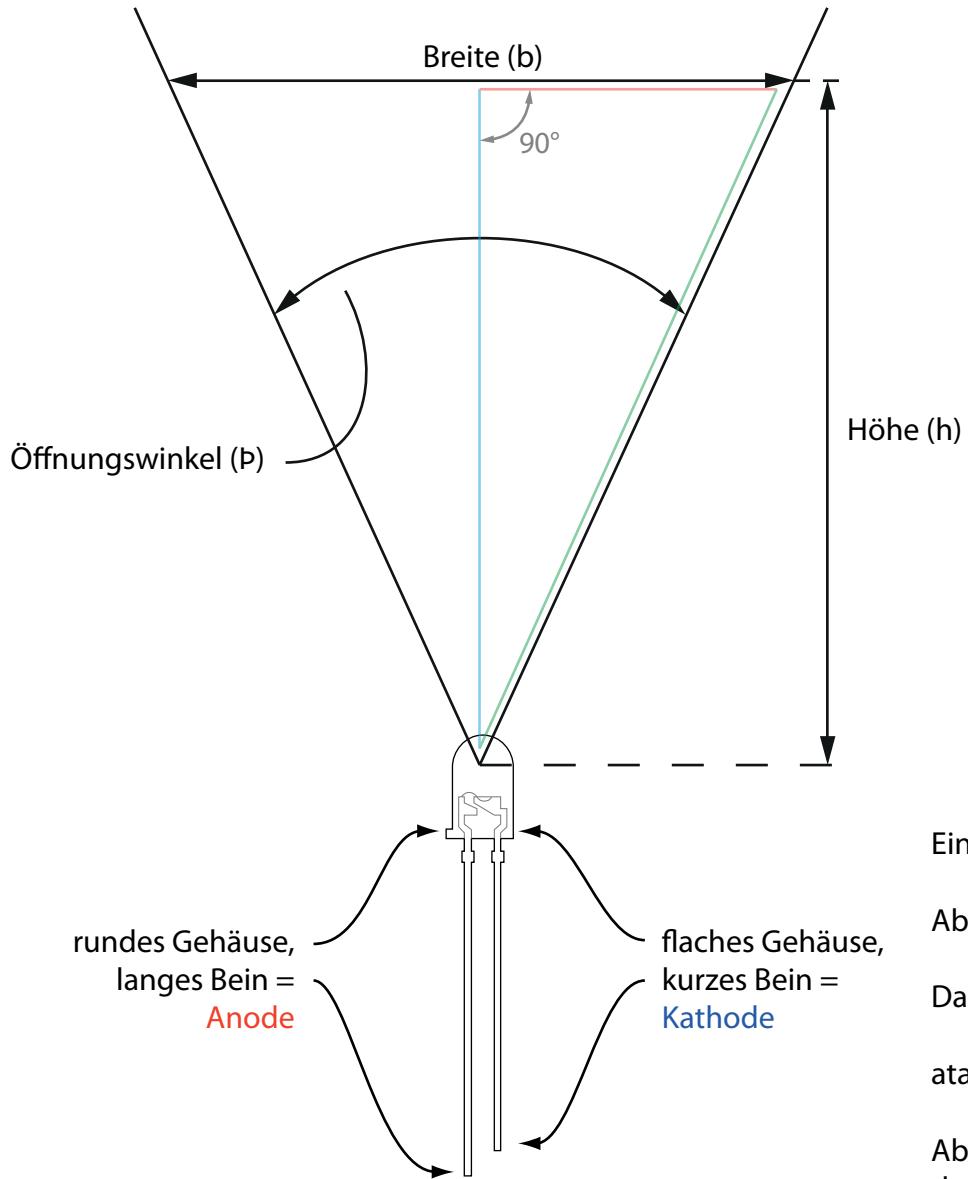
Das ist eine Superflux-LED. Sie ist klein, flach und hat neben einer enormen Helligkeit einen Abstrahlwinkel von 120°. Sie ist schwieriger zu verarbeiten, aber man könnte ein flacheres und damit eleganteres Gehäuse bauen.



Das ist eine RGB-Pixel von bliptronics. Er kann Licht in 32.768 Farben abgeben! Und das mit einem Abstrahlwinkel von 160°! Man stelle sich vor: die Matrix kann die Farben wechseln, oder zwischen den Wörtern überblenden!



Appendix C - Der Öffnungswinkel



$$\sin(\alpha) = \frac{\text{Gegenkathete}}{\text{Hypotenuse}} = \frac{a}{c}$$

$$\cos(\alpha) = \frac{\text{Ankathete}}{\text{Hypotenuse}} = \frac{b}{c}$$

$$\tan(\alpha) = \frac{\text{Gegenkathete}}{\text{Ankathete}} = \frac{a}{b}$$

Ein Kästchen hat 35mm Kantenlänge, damit $a = 17,5\text{mm}$.

Abstand Bodenplatte zu Glas 25mm, damit $b = 25\text{mm}$.

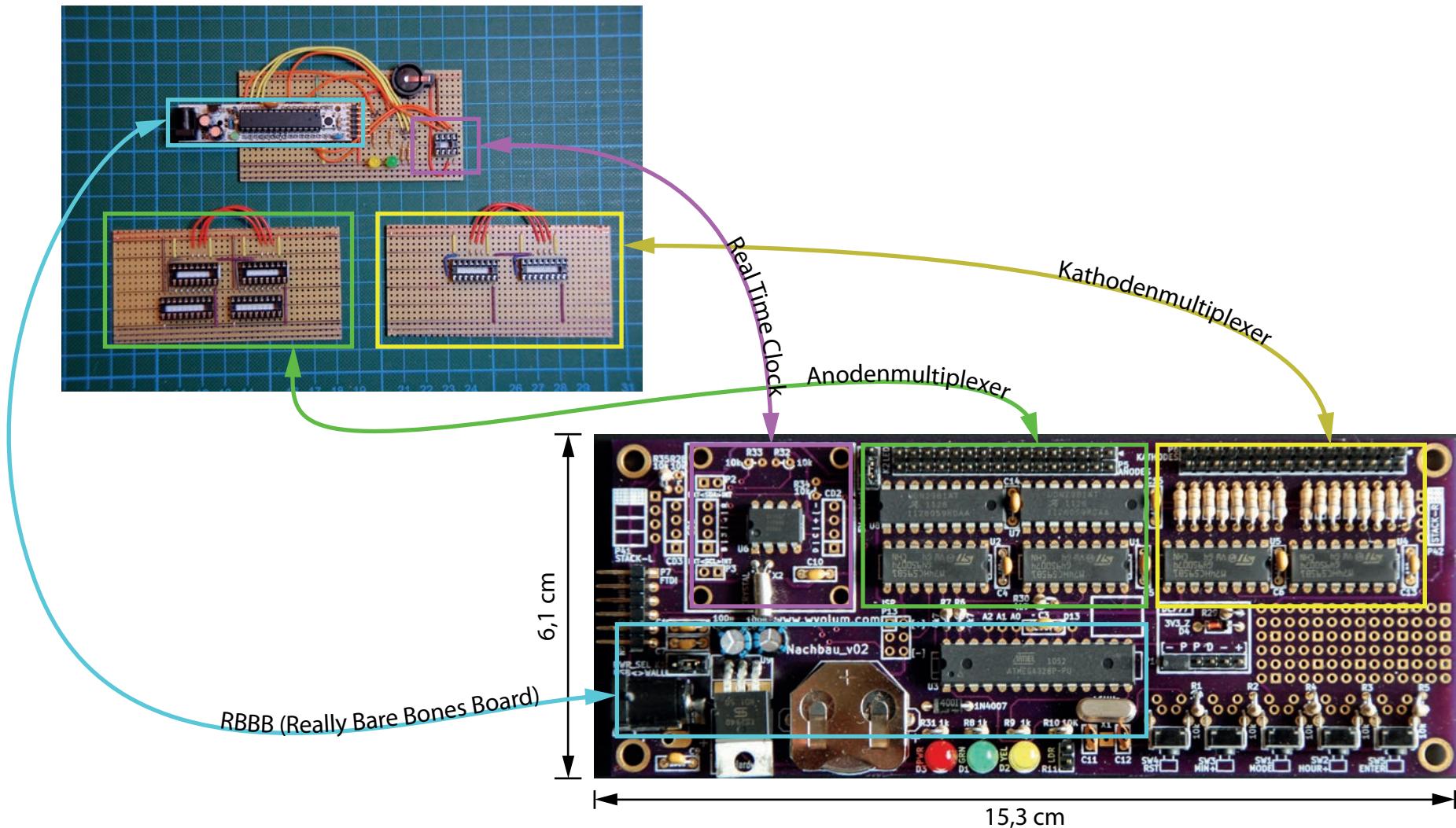
Damit $\tan(\alpha) = 17,5\text{mm}/25\text{mm} = 0,7$.

$\text{atan}(0,7) = 35^\circ$, da Öffnungswinkel (β) = $2 \cdot \alpha$ ist $\beta \text{ mind. } 70^\circ$.

Aber Achtung! Viele Hersteller geben als Öffnungswinkel den Winkel an, bei dem die Helligkeit auf 50% bzw. 0° abfällt. Außerdem sind die Einheit Lumen und (Milli-) Candela schwer zu vergleichen. Googeln nach „lumen millicandela converter“ lohnt sich.

Appendix D - Die NachBau_V02-Platine

Justin (aus Washington DC, USA) und Anool (aus Mumbai, Indien) haben eine Platine entworfen, die alle drei Steuerplatinen und das RBBC in sich vereint. Dadurch wird natürlich das Kabel-Chaos stark reduziert und außerdem sinkt die Fehlerrate ganz enorm. Aber man ist auch nicht mehr so „nah“ an der Technik. Diese Platine kann man alleine oder auch mit allen Bauteilen unter „<http://wyolum.com/shop/6-nachbau>“ bestellen. Eine Beschreibung und Aufbauanleitung gibt es bei den Downloads unter „<http://code.google.com/p/grosses-bastelwochenende/>“.



Appendix E - Hacking the Firmware...

Die Firmware des Nachbauprojektes ist dazu gedacht, von Jedermann mit Interesse „gehackt“ zu werden, also so verändert zu werden, daß sie den eigenen Vorstellungen und Bedürfnissen entspricht. Andreas und Benedikt haben das z.B. schon erfolgreich mit ihrer „Multiqlock“ gemacht, die auch die Temperatur und anderes anzeigen kann. Dazu möchte ich hier die wichtigsten Hinweise geben.

Übersicht über die Dateien (Version 2.0):

Qlockthree.ino (früher .pde):	Das Herz der Firmware, hier wird in setup() die Initialisierung vorgenommen und in loop() die Software ausgeführt. Relativ am Anfang gibt es viele #define-Compiler-Schalter, mit welchen das Verhalten gesteuert werden kann.
Button (.cpp und .h):	Kleine Hilfsklasse für das Entprellen von Tastern. Hier kann die Wiederholungsrate der Taster verändert werden.
DCF77Helper (.cpp und .h):	Hilfsklasse um Telegramme des DCF77-Empfängers zu überprüfen. Dazu merkt sich die Klasse die Zeit von der DS1307-Echtzeituhr bei jedem als gültig erkannten DCF77-Telegramm. Dann wird verglichen, ob der zeitliche Abstand der Echtzeituhr der gleiche Abstand ist, wie die DCF77-Telegramme. Stimmen drei Abstände, wird das DCF77-Telegramm als korrekt akzeptiert und die DS1307-Echtzeituhr danach gestellt.
DS1307 (.cpp und .h):	Hilfsklasse für das Auslesen und Stellen der Echtzeituhr.
LDR (.cpp und .h):	Hilfsklasse um die Werte des LDR zu glätten und die Grenzwerte automatisch zu ermitteln. Hier kann an der Helligkeit gedreht werden.
MyDCF77 (.cpp und .h):	Hilfsklasse für das Empfangen von DCF77-Telegrammen.
ShiftRegister (.cpp und .h):	Hilfsklasse für das Schreiben auf die 74HC595 Shift Register.
TimeStamp (.cpp und .h):	Hilfsklasse für die verschiedenen Zeitstempel (DS1307 vs. DCF77).

Potentiell **sprachabhängige** Dateien (dann mit Anhängsel wie **_DE** (für Deutsch) oder **_CH** (für Schweizerisch)), in der Datei „Qlockthree.ino“ werden dann die Dateien inkludiert, die der gewünschten Sprache entsprechen:

SetCorners (.h):	Setzt die Eck-LEDs, hier kann bei falscher Verdrahtung der Eck-LEDs geschraubt werden. Dazu muss man die Reihenfolg der Zahlen in den eckigen Klammern tauschen (aber es bleibt bei den Zahlen 0-3, die die Reihe der Matrix angeben, an die man die Anoden angeloetet hat).
SetHours (.h):	Setzt die Stunden.
SetMinutes (.h):	Setzt die Minuten.
Woerter (.h):	Definition der Wörter in der Matrix, muß mit dem Folienlayout übereinstimmen.
Zahlen (.h):	Definition der Zahlen für die Sekundenanzeige.

Hacking the Firmware... (fortgesetzt)

Compilerschalter

Compilerschalter werden ein- oder ausgeschaltet, indem man sie ein- oder auskommentiert. Bei Compilerschaltern mit einem Wert wird einfach der Wert verändert.

```
#define EINGESCHALTET      einkommentiert  
// #define AUSGESCHALTET   // auskommentiert
```

(Qlockthree.ino):

```
#define DEBUG  
  
#define SERIAL_SPEED 57600  
  
#define ENABLE_DCF_LED  
#define ENABLE_SQW_LED  
#define ENABLE_LDR  
#define SKIP_BLANK_LINES  
#define UPSIDE_DOWN  
#define SPLIT_SIDE_DOWN
```

Schaltet Debug-Ausgaben ein. Diese kann man sehen, wenn man den Serial Monitor in der Arduino-IDE einschaltet.
Einen #debug-Schalter gibt es in fast jeder Klasse.
Geschwindigkeit der Ausgabe der Debug-Ausgaben. Im Serial Monitor muß die gleiche Geschwindigkeit gewählt sein.
Soll die DCF77-Status-LED eingeschaltet sein?
Soll die DS1307-Status-LED eingeschaltet sein?
Soll die Helligkeit mit einem lichtabhängigen Widerstand automatisch angepasst werden?
Sollen Zeilen, in denen keine LED leuchtet, übersprungen werden? Dann werden die restlichen LEDs heller.
Soll die Anzeige auf den Kopf gestellt werden? (Kabel beim Anoden-Multiplexer nicht gedreht?)
Sollen die Shift Register des Anodenmultiplexers getauscht werden?

(Button.cpp):

```
#define BUTTON_THRESHOLD 300
```

Ansprechverzögerung für die Taster.

(DCF77Helper.h):

```
#define DCF77HELPER_MAX_SAMPLES 3
```

Anzahl der Telegramme, die im richtigen Abstand gekommen sind, bevor das Telegramm als gültig in die DS1307-Echtzeituhr geschrieben wird.

(LDR.h):

```
#define LDR_AUTOSCALE  
#define MAX_BRIGHTNESS 9  
#define LDR_MEAN_COUNT 32
```

Sollen die Grenzwerte des LDRs automatisch angepasst werden?
Maximale Helligkeit, auf welche die Werte des LDRs skaliert werden.
Anzahl der Messwerte, über die der gleitende Mittelwert gebildet wird.

Hacking the Firmware... (fortgesetzt)

Compilerschalter (fortgesetzt)

(MyDCF77.h):

```
#define MYDCF77_SYNC_PAUSE_THRESHOLD 1700
```

Wartezeit in Millisekunden, nach denen davon ausgegangen wird, daß ein neues Telegramm beginnt (1 Sekunde + Zeit bis zum nächsten Anstieg des ersten Bits).

```
#define MYDCF77_200MS_THRESHOLD 185
```

Wartezeit in Millisekunden, nach denen davon ausgegangen wird, daß eine „!“ empfangen wurde.

```
#define MYDCF77_100MS_THRESHOLD 85
```

Wartezeit in Millisekunden, nach denen davon ausgegangen wird, daß eine „0“ empfangen wurde.

```
#define MYDCF77_TELEGRAMMLAENGE 59
```

Telegrammlänge, dieser Wert sollte nie geändert werden.

(ShiftRegister.h):

```
#define SHIFTREGISTER_TURBO
```

Dieser Schalter beschleunigt das Schreiben auf die Shift Register (74HC595), in dem verschiedene Prüfungen der Arduino-IDE nur einmalig und nicht bei jedem Scheiben durchgeführt werden.

Ändern der Sprache

(Qlockthree.ino):

Die Wörter definieren die Positionen der Wörter auf der Matrix. Neue Sprachen benötigen meistens neue Vorlagen für den Folienplot.

```
#include "Woerter_DE.h" // Deutsch, auch Schwäbisch
```

```
#include "Woerter_CH.h" // Schweizerdeutsch
```

```
#include "SetMinutes_DE_DE.h" // Hochdeutsch
```

```
#include "SetMinutes_DE_SW.h" // Schwäbisch (viertel Drei; dreiviertel Zehn):
```

```
#include "SetMinutes_CH.h" // Schweizerdeutsch
```

```
#include "SetHours_DE.h" // Deutsch, auch Schwäbisch
```

```
#include "SetHours_CH.h" // Schweizerdeutsch
```