

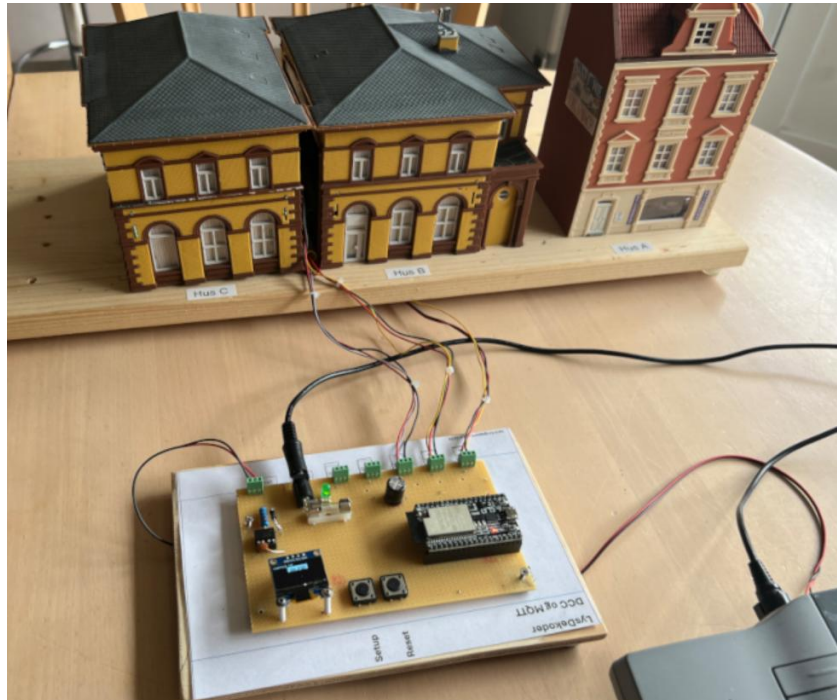
Selbstgebauter Lichtdecoder mit DCC und MQTT

Dokument erstellt 09-06-2026

Rev 0: erstellt

Rev 1: 14-06-2026 : Die Konfiguration wurde um Firmware und Hostname erweitert. Der Screenshot wurde entsprechend aktualisiert.

Jens Krogsgaard, jenskrogsgaard@gmail.com, Syrenvænget 10, 3520 Farum, Denmark - +4542669987



Indhold

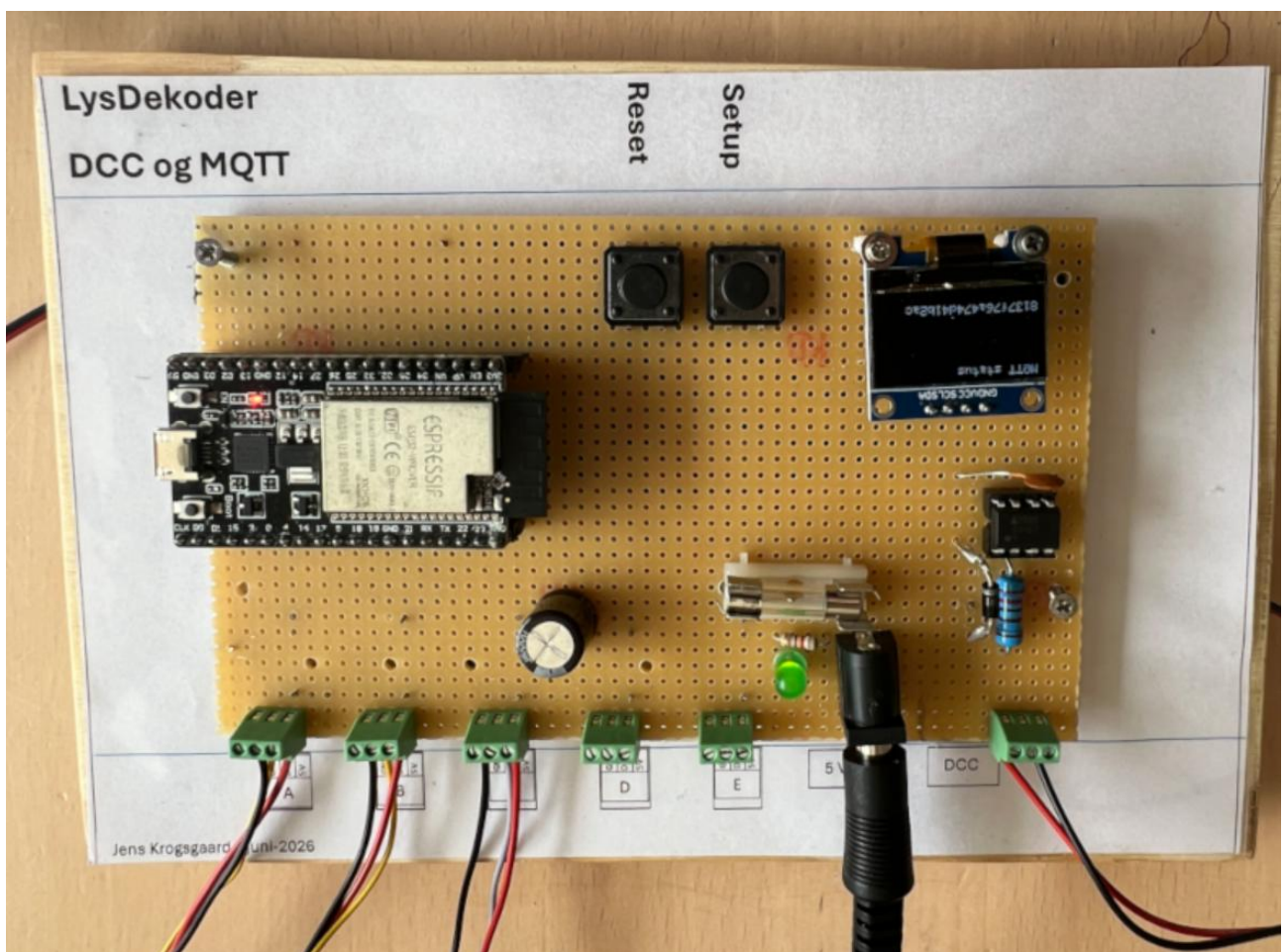
1	Zusammenfassung	2
2	Was kann der Decoder?	3
3	Anschlüsse der Decoders	3
4	Erste Inbetriebnahme	4
5	Das kleine Display	6
6	Konfiguration	7
7	Technik	13
8	Programmierung des ESP32	14
9	Windigipet 2025	15

1 Zusammenfassung

Auf einer Modellbahn sieht es meiner Meinung nach viel realistischer aus, wenn die Lichter in einem Gebäude nicht alle gleichzeitig eingeschaltet werden, sondern in den einzelnen Räumen zufällig ein- und ausgeschaltet werden.

Früher habe ich das mit normalen LEDs und einem M84/K84-Decoder realisiert. Diese Decoder sind jedoch recht teuer, deshalb habe ich nach alternativen Möglichkeiten gesucht.

Nun habe ich einen Decoder entwickelt, der sowohl DCC als auch MQTT unterstützt. DCC kennen die meisten Modellbahner, und praktisch jede Digitalzentrale unterstützt dieses Protokoll. MQTT wird zwar seltener verwendet, aber beispielsweise Win-Digipet 2025 nutzt es bereits. Warum also nicht einen Decoder bauen, der beides beherrscht – entweder gleichzeitig oder unabhängig voneinander?



2 Was kann der Decoder?

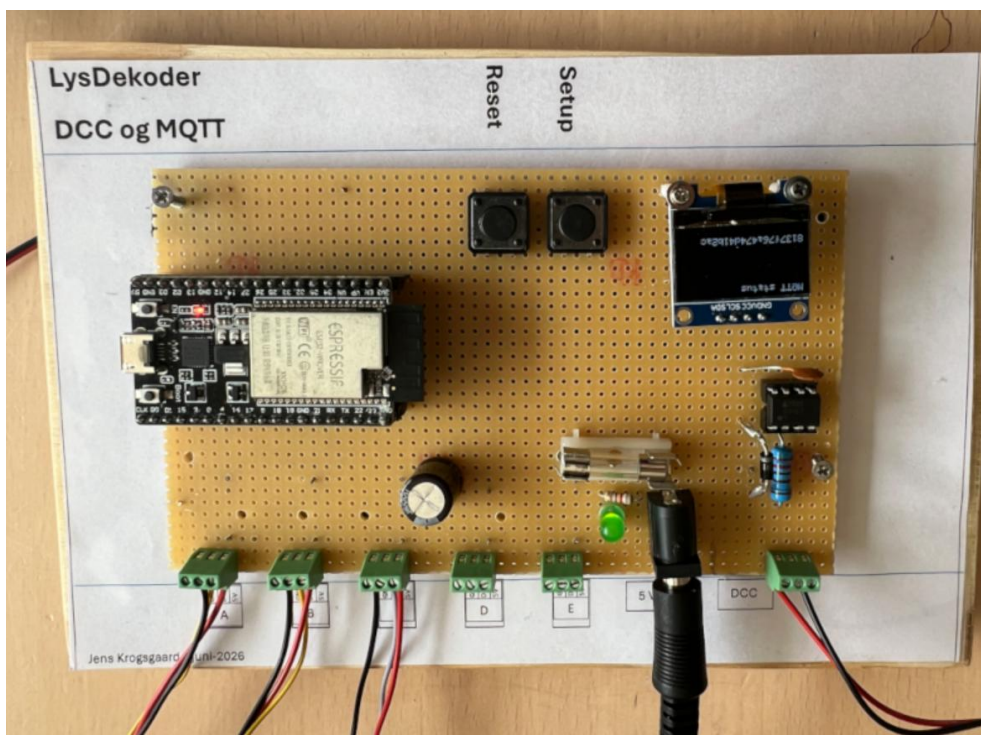
Der Decoder kann bis zu fünf programmierbare LED-Streifen steuern. Diese laufen meist unter der Bezeichnung WS2812B und sind in unzähligen Varianten erhältlich – unterschiedliche Längen, LED-Anzahlen usw.



Für den Einsatz auf einer Modellbahn eignen sich diese LEDs hervorragend. Ein Streifen mit beispielsweise 20 LEDs kann in einem Gebäude eingebaut werden, wobei jede LED einzeln gesteuert werden kann.

Jede LED erhält ihre eigene Adresse und kann mit meinem Decoder individuell hinsichtlich Farbe, Helligkeit und Effekt konfiguriert werden.

3 Anschlüsse der Decoders



Der Decoder verfügt über folgende Anschlüsse:

- A–E: Anschluss für bis zu fünf LED-Streifen
- 3 Leitungen pro Streifen:
 - GND
 - Data
 - 5 V
- 5 Volt DC Versorgung
 - Hier kann beispielsweise ein altes Handy-Ladegerät verwendet werden
- DCC-Eingang
 - Anschluss an die Digitalzentrale (nur erforderlich, wenn DCC verwendet wird)

Sobald die 5-Volt-Versorgung angeschlossen wird, leuchtet die grüne LED auf. Anschließend kann man auf dem kleinen Display verfolgen, was der Decoder gerade macht.

4 Erste Inbetriebnahme.

Wenn der Decoder zum ersten Mal eingeschaltet wird, muss er mit dem WLAN verbunden werden



Kennt er das WLAN noch nicht, kann die Verbindung ganz einfach mit dem Smartphone eingerichtet werden.

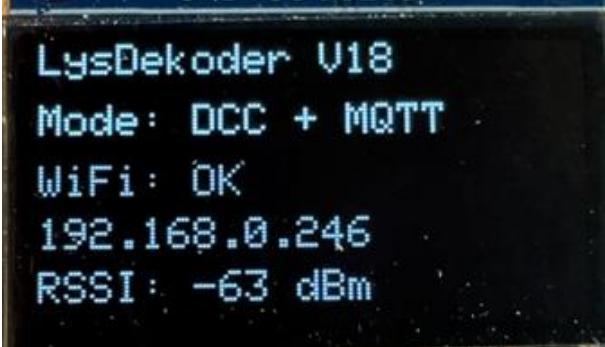
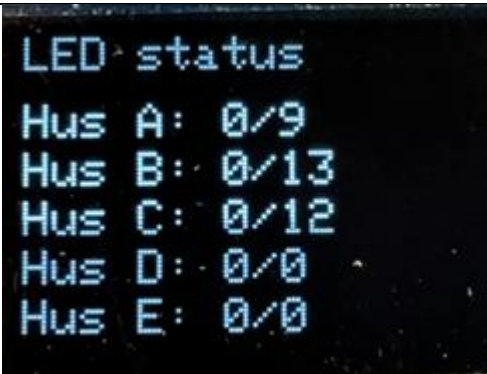
Verbinde dich mit dem angezeigten Netzwerk. Danach öffnet sich eine Seite, auf der du dein WLAN auswählen und das Passwort eingeben kannst.



Anschließend verbindet sich der Decoder mit dem Netzwerk und speichert die Zugangsdaten für zukünftige Starts.

5 Das kleine Display 😊

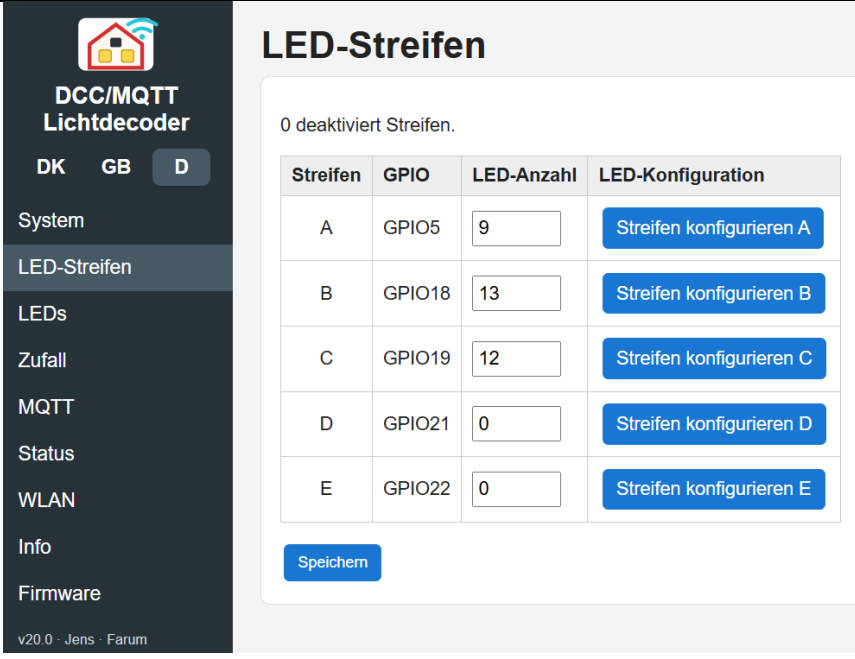
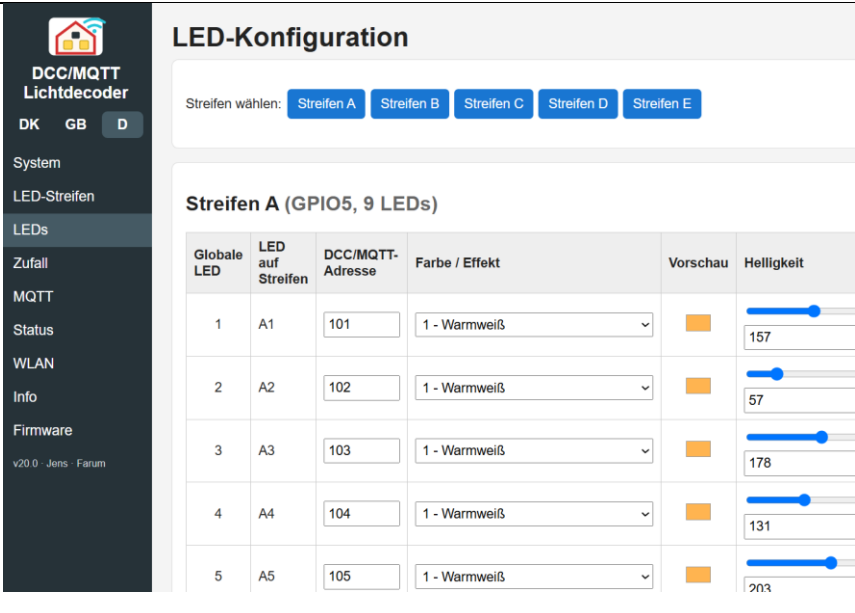
Das Display des Decoders zeigt verschiedene Informationen an. Zwischen den einzelnen Anzeigen wird automatisch alle paar Sekunden umgeschaltet.

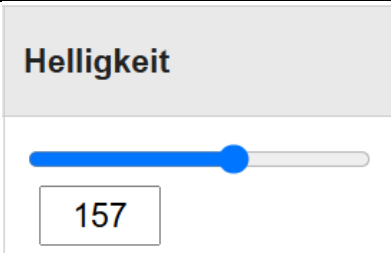
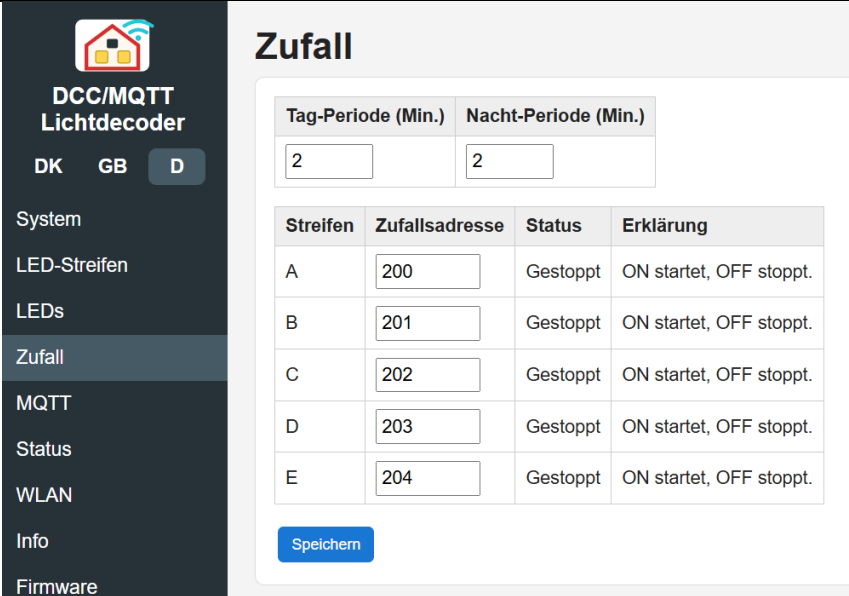
Screenshot	Erläuterung
	Das WLAN wird geprüft – hier ist alles in Ordnung. Außerdem wird die Adresse der Konfigurationsseite angezeigt. In diesem Beispiel: 192.168.0.246
	Hier kann man sehen, dass die Verbindung zum MQTT-Broker erfolgreich hergestellt wurde.
	Hier wird angezeigt, welche der fünf LED-Streifen verwendet werden, wie viele LEDs sie enthalten und wie viele davon aktuell eingeschaltet sind.
	Hier wird ein QR-Code für die Konfigurationsseite angezeigt. Einfach mit dem Smartphone scannen und die Konfigurationsseite wird direkt geöffnet. Alternativ kann die im ersten Bildschirm angezeigte IP-Adresse auch manuell eingegeben werden.

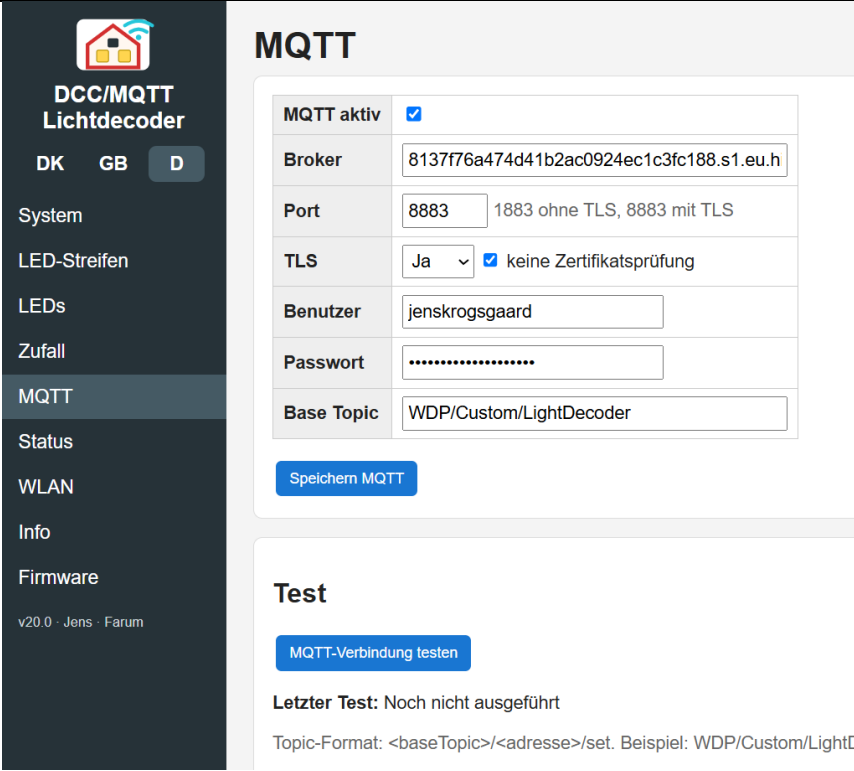
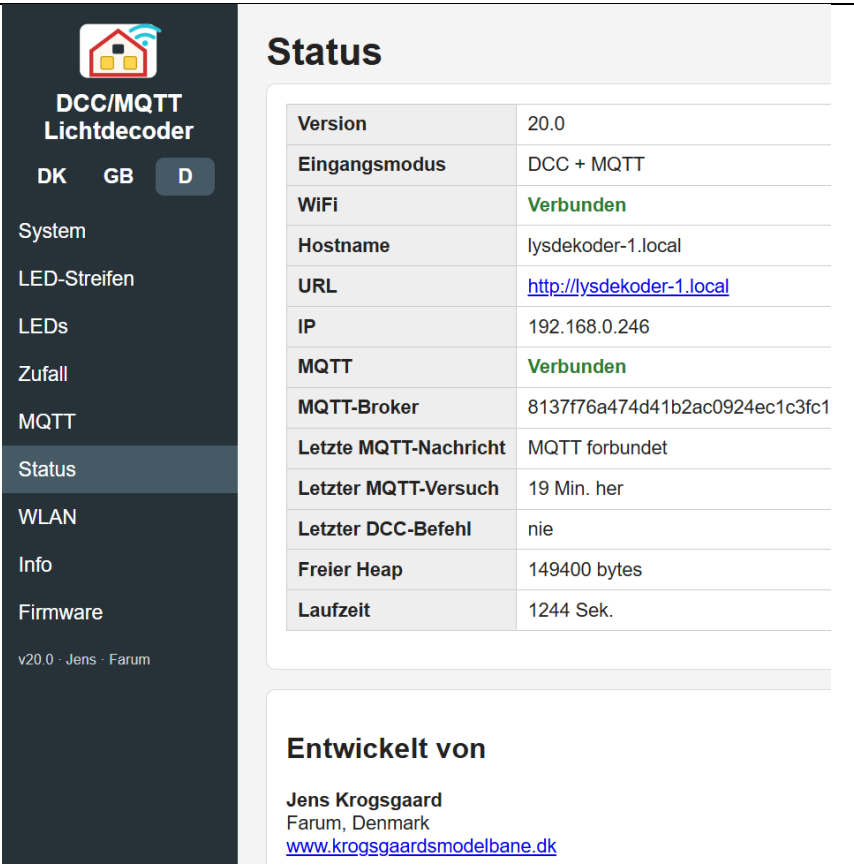
6 Konfiguration

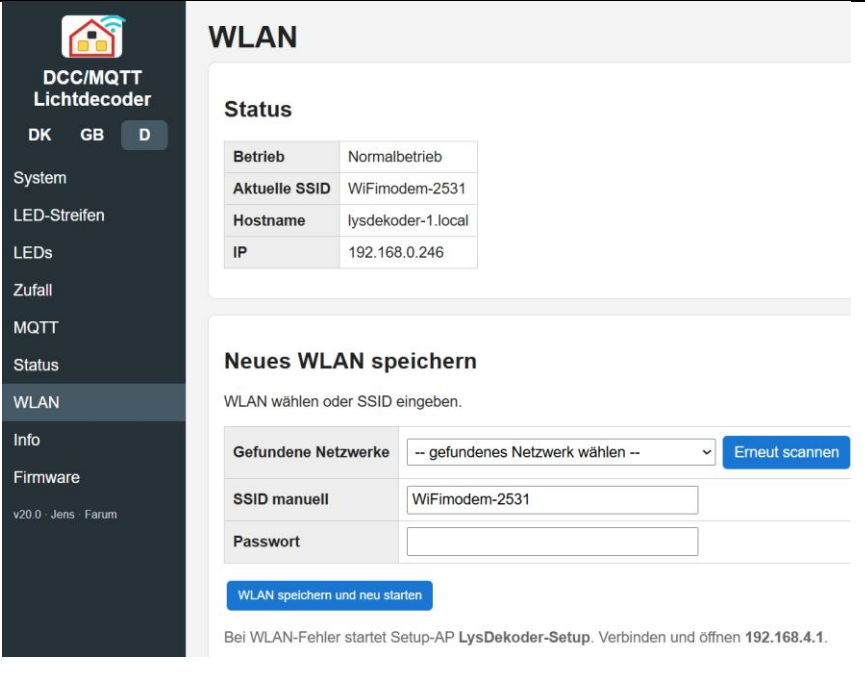
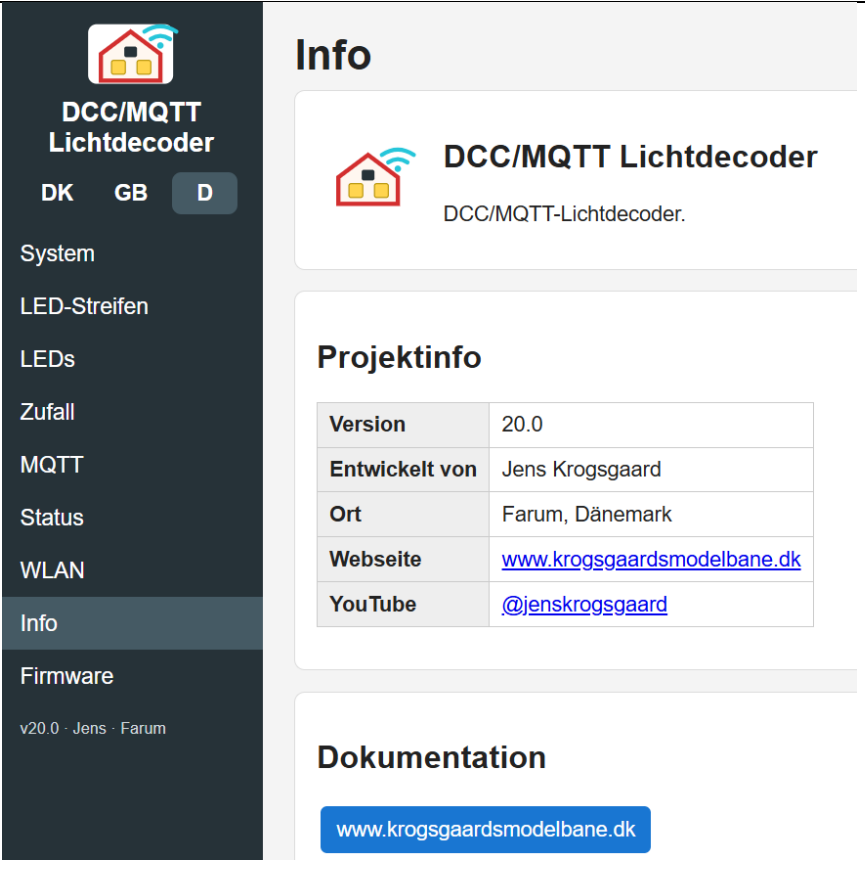
Durch Eingabe der Adresse oder Scannen des QR-Codes gelangt man auf die Konfigurationsseite des Decoders.

Screenshot	Erläuterung
	Startseite Hier kann ausgewählt werden, ob DCC und/oder MQTT verwendet werden sollen. Es ist außerdem möglich, einen Hostnamen festzulegen – praktisch, wenn sich die IP-Adresse ändert.
	Hier kann die gewünschte Sprache ausgewählt werden. Alle Texte der Benutzeroberfläche werden sofort entsprechend angepasst.

Screenshot	Erläuterung
	Es können bis zu fünf LED-Streifen angeschlossen werden. Hier wird die Anzahl der LEDs jedes verwendeten Streifens eingetragen.
	Hier werden die einzelnen LEDs konfiguriert. Jeder LED kann eine eigene Adresse zugewiesen werden oder man übernimmt die vorgeschlagenen Adressen. Grundsätzlich können beliebige Adressen verwendet werden; sie müssen nicht in einem zusammenhängenden Bereich liegen. Aus Gründen der Übersichtlichkeit empfiehlt sich jedoch eine strukturierte Vergabe.
Feste Farben 1 - Warmweiß 2 - Kaltweiß 3 - Rot 4 - Grün 5 - Blau 6 - Gelb 7 - Orange 8 - Lila 9 - Cyan 10 - Gedimmtes Nachtlicht Effekte 11 - TV-Flimmern 12 - Schweißlicht 13 - Feuerflackern 14 - Gedimmtes Schlafzimmer 15 - Defekte Neonröhre 16 - Straßenlampe / weicher Puls 17 - Zufälliges Hauslicht 18 - Warnblinken 19 - Discolicht	Für jede LED kann entweder eine Farbe oder ein Effekt ausgewählt werden.

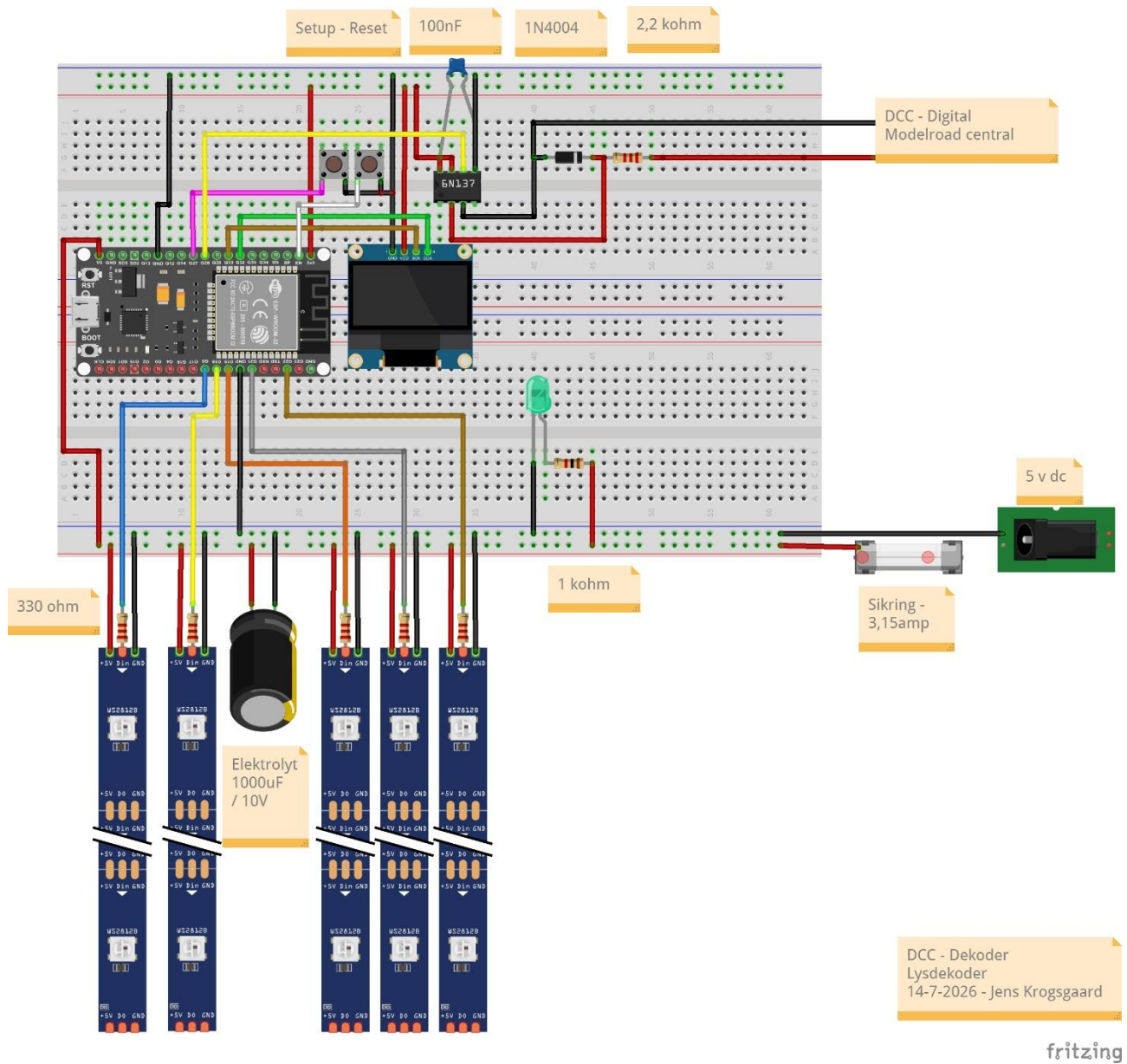
Screenshot	Erläuterung
	Die Helligkeit kann für jede einzelne LED individuell eingestellt werden.
	Jeder LED-Streifen besitzt eine Random-Funktion, die über eine frei wählbare Adresse aktiviert werden kann. Zusätzlich wird festgelegt, wie viele Minuten Tag und Nacht dauern sollen. Wird die Random-Funktion aktiviert, beginnt zunächst die Nacht. Nach und nach werden einzelne LEDs eingeschaltet. Wenn der Morgen näher rückt, gehen die LEDs ebenfalls nach und nach wieder aus. Dadurch entsteht ein sehr realistischer Eindruck von belebten Gebäuden.

Screenshot	Erläuterung
	Hier können die Verbindungsdaten zum MQTT-Broker eingetragen und getestet werden.
	Hier wird der aktuelle Status des Systems angezeigt. Man kann sofort erkennen, ob WLAN und MQTT verbunden sind.

Screenshot	Erläuterung
	Hier können die gespeicherten WLAN-Daten angezeigt und bei Bedarf geändert werden.
	Hier finden sich allgemeine Informationen zum System, zur Softwareversion und zum Entwickler.

Screenshot	Erläuterung
 The screenshot shows a web interface for a DCC/MQTT Lichtdecoder. On the left is a dark sidebar with a home icon and menu items: DCC/MQTT Lichtdecoder, DK, GB, D (selected), System, LED-Streifen, LEDs, Zufall, MQTT, Status, WLAN, Info, Firmware (highlighted), and v20.0 · Jens · Forum. The main content area is titled 'Firmware-Update' and contains a section 'Neue Firmware hochladen' with instructions to upload ESP32-Firmware (.ino.bin), a file selection button labeled 'Vælg fil' with a warning 'Der er ikke valgt nogen fil', and a blue button 'Hochladen und neu starten'. Below this is a 'Wichtig' (Important) section with the text 'Richtige Firmware. Nicht ausschalten.' and 'Aktuelle Version: 20.0'.	Hier besteht die Möglichkeit, eine neue Firmware auf den ESP32 oder den NodeMCU hochzuladen. Das ist besonders praktisch, da einige Geräte an Stellen eingebaut sind, die nur schwer zugänglich sind. Die Firmware-Datei wird in der Arduino IDE über den Menüpunkt <i>Sketch</i> → <i>Export Compiled Binary</i> erzeugt.

7 Technik



Komponenten:

Komponent	Link
ESP32	https://ardustore.dk/produkt/nodemcu-esp32-wrover-38pin-cp2102-udviklingsboard
Elektrolytkondensator	https://www.elextra.dk/p/470uf-16v-lodret-elektrolyt-%C3%B88x12mm-105c/H39614
5-Volt-Gleichspannungsversorgung	Altes Ladegerät eines Mobiltelefons
Optokobler 6N137	https://www.elextra.dk/p/6n137-optokobler-m-transistorudgang-1-kan-dip8/H15808
Widerstände	300 ohm, 2.2 kOhm, 1 kOhm
Lodret elektrolyt	https://www.elextra.dk/p/1000uf-10v-lodret-elektrolyt-%c3%b810x13mm-105c/H32763
Kondensator	https://www.elextra.dk/p/keramisk-multilag-kondensator-100nf-01uf-100v-508mm/H12203
Sicherung – 3,15 Amp	https://www.elextra.dk/p/sikring-%C3%B85x20mm-tr%C3%A6g-315a/H15293
Oled display	https://www.elextra.dk/p/096-oled-display-m-i2c/H59880
Taster	https://www.elextra.dk/p/tact-mikrotrykkontakt-12x12mm-h43mm/H12480
Gleichrichterdiode	https://www.elextra.dk/search?q=1N4004
LED	Grün Led

Zusätzlich zu den aufgeführten Bauteilen habe ich Sockel für ESP32, OLED-Display und Optokoppler, eine Lochrasterplatine sowie verschiedene Steckverbinder und Schraubklemmen verwendet.

Für Tests ist ein programmierbarer LED-Streifen wie beispielsweise ein WS2812B-LED-Streifen sehr praktisch.

<https://www.elextra.dk/p/argb-led-strip-til-pc-5v-ws2812b-3p-50cm/H54986>

Die Komponenten werden gemäß dem Schaltplan aufgebaut. Anschließend kann die Programmierung des ESP32 beginnen.

8 Programmierung des ESP32.

Die Software wurde in enger Zusammenarbeit mit ChatGPT entwickelt.

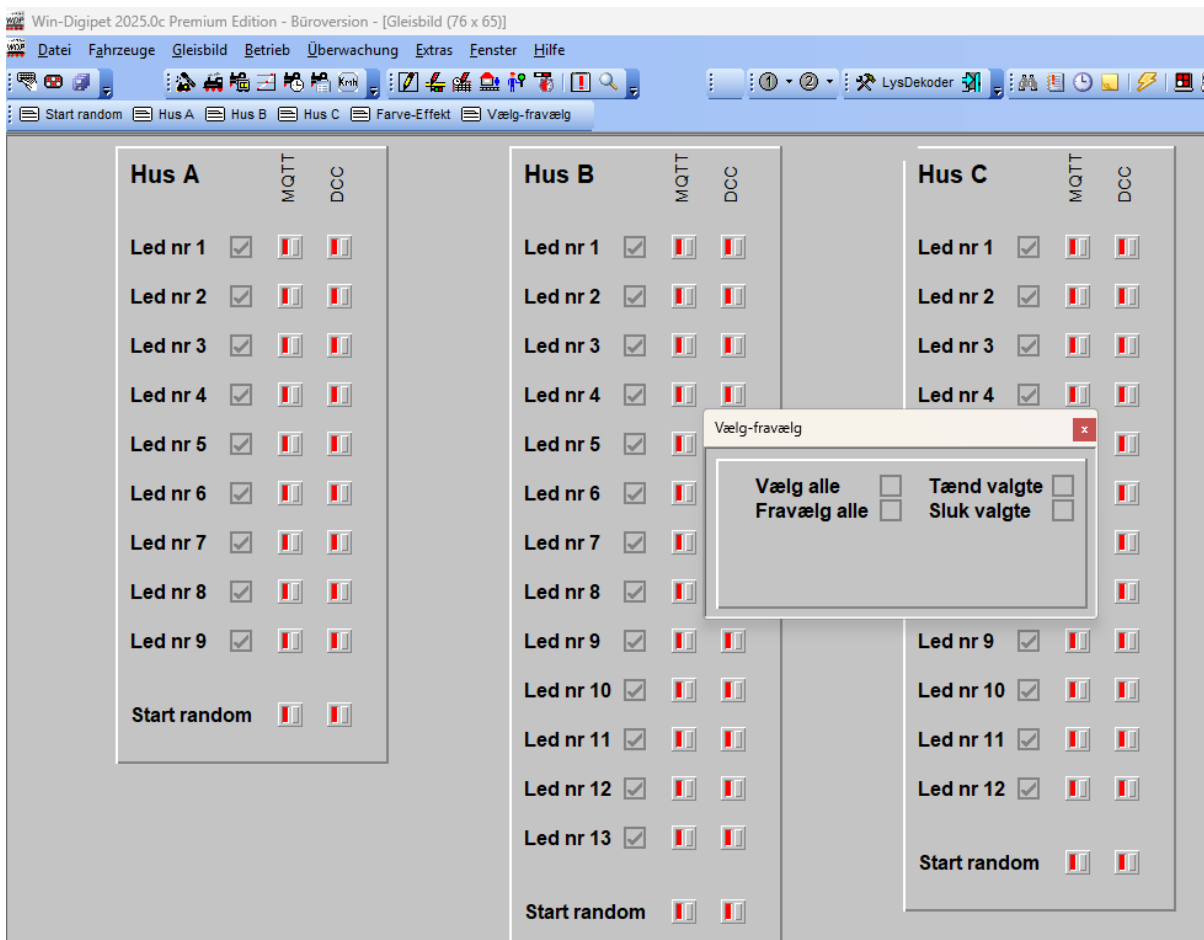
Die Arbeitsteilung war einfach: Ich habe die Anforderungen beschrieben, und ChatGPT hat den Code geschrieben.

Die Zusammenarbeit war ausgesprochen angenehm – sowohl beim gemeinsamen Entwickeln von Ideen als auch bei der schnellen Erstellung von Programmcode, der meist auf Anhieb funktioniert hat.

Wenn du Interesse hast, kannst du mich gerne kontaktieren. Ich stelle den Quellcode gerne zur Verfügung.

Als Entwicklungsumgebung habe ich die Arduino IDE verwendet.

9 Windigipet 2025

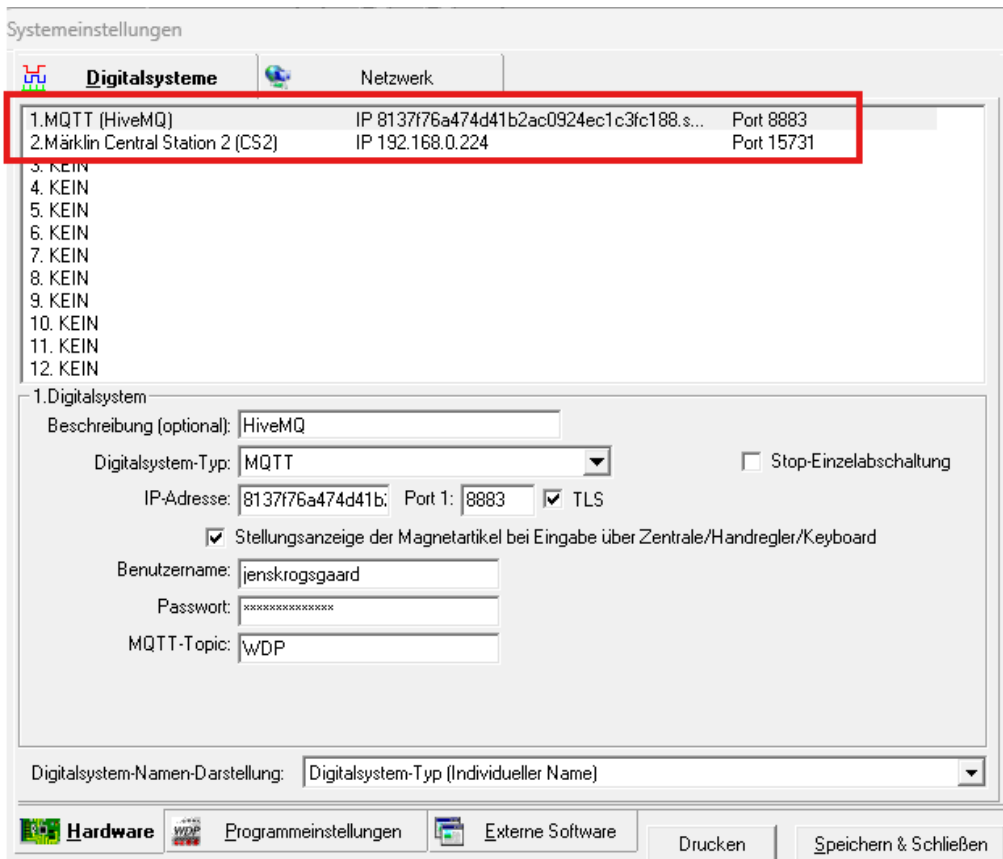


In Win-Digipet 2025 habe ich eine kleine Testseite eingerichtet, mit der ich sowohl MQTT als auch DCC ausprobieren kann.

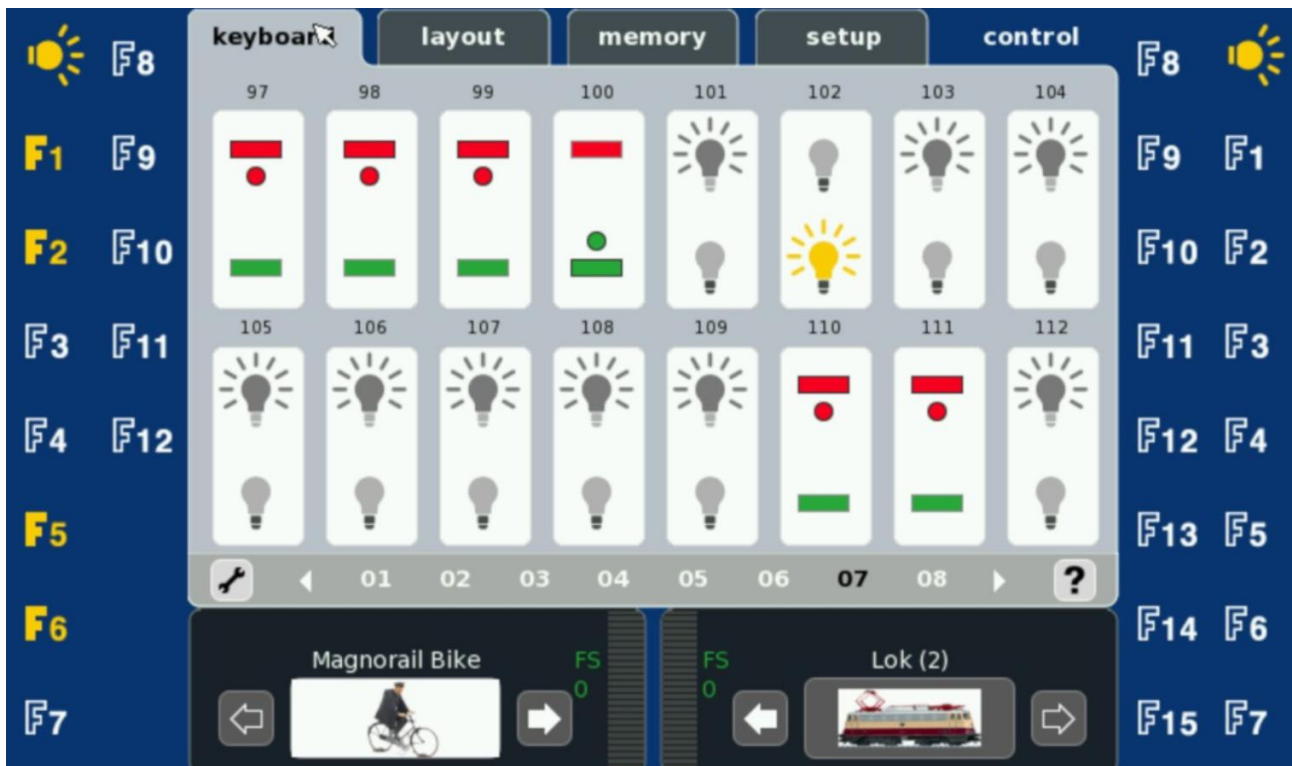
Ich habe dort drei Häuser, die jeweils mit einem eigenen LED-Streifen ausgestattet sind. Jede einzelne LED kann ich entweder über MQTT oder über DCC ein- und ausschalten.

Das ist eine sehr praktische Möglichkeit, das gesamte System zu testen.

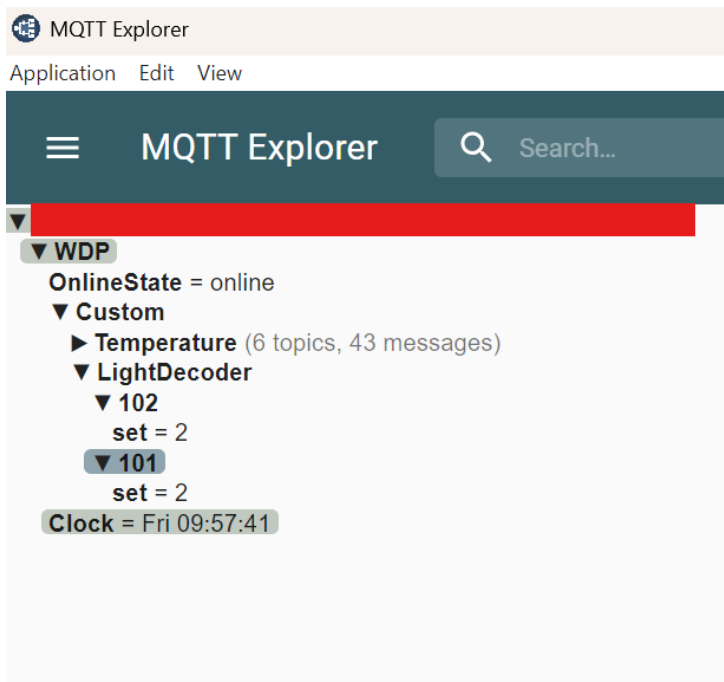
Besonders interessant ist, dass MQTT genauso schnell reagiert wie DCC. Einen Unterschied in der Reaktionszeit kann man in der Praxis kaum feststellen.



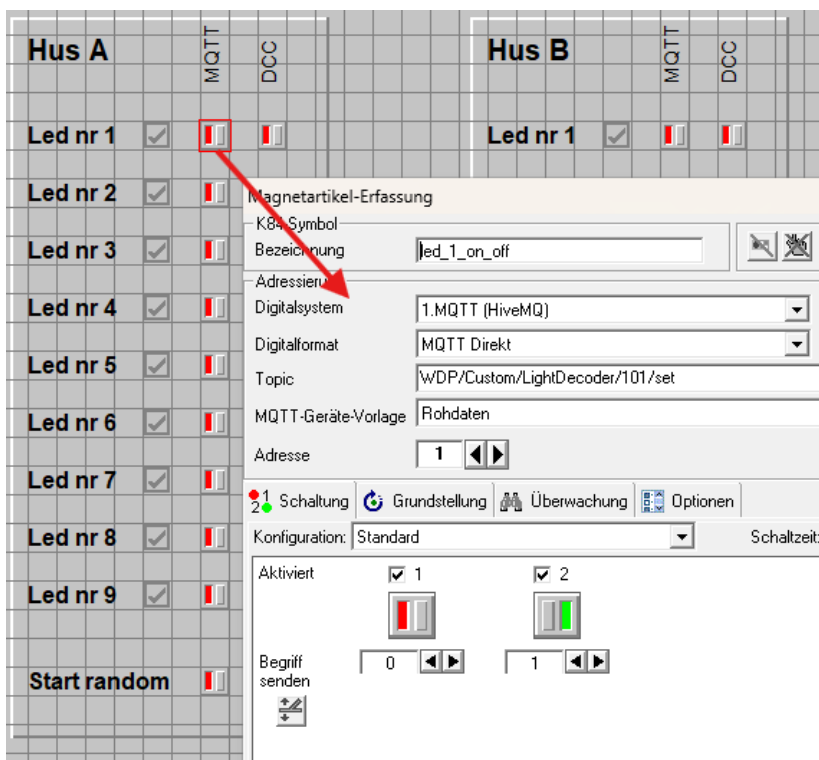
In den Systemeinstellungen habe ich zwei Digitalsysteme definiert – meine alte CS2 und MQTT.



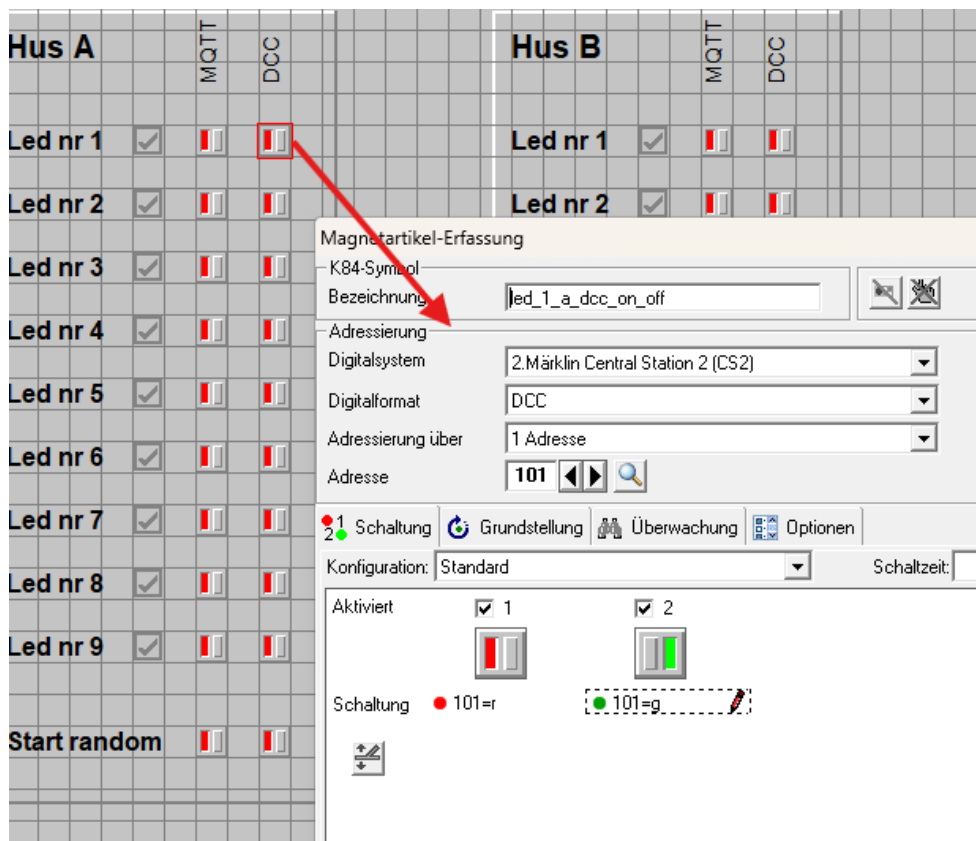
DCC kann natürlich auch direkt über die Digitalzentrale getestet werden – in meinem Fall über meine alte CS2.



MQTT kann mit dem MQTT Explorer getestet werden.



In WinDigipet definiere ich die MQTT-Schaltfläche wie folgt



Und die DCC-Schaltfläche wird ganz wie gewohnt definiert. 😊