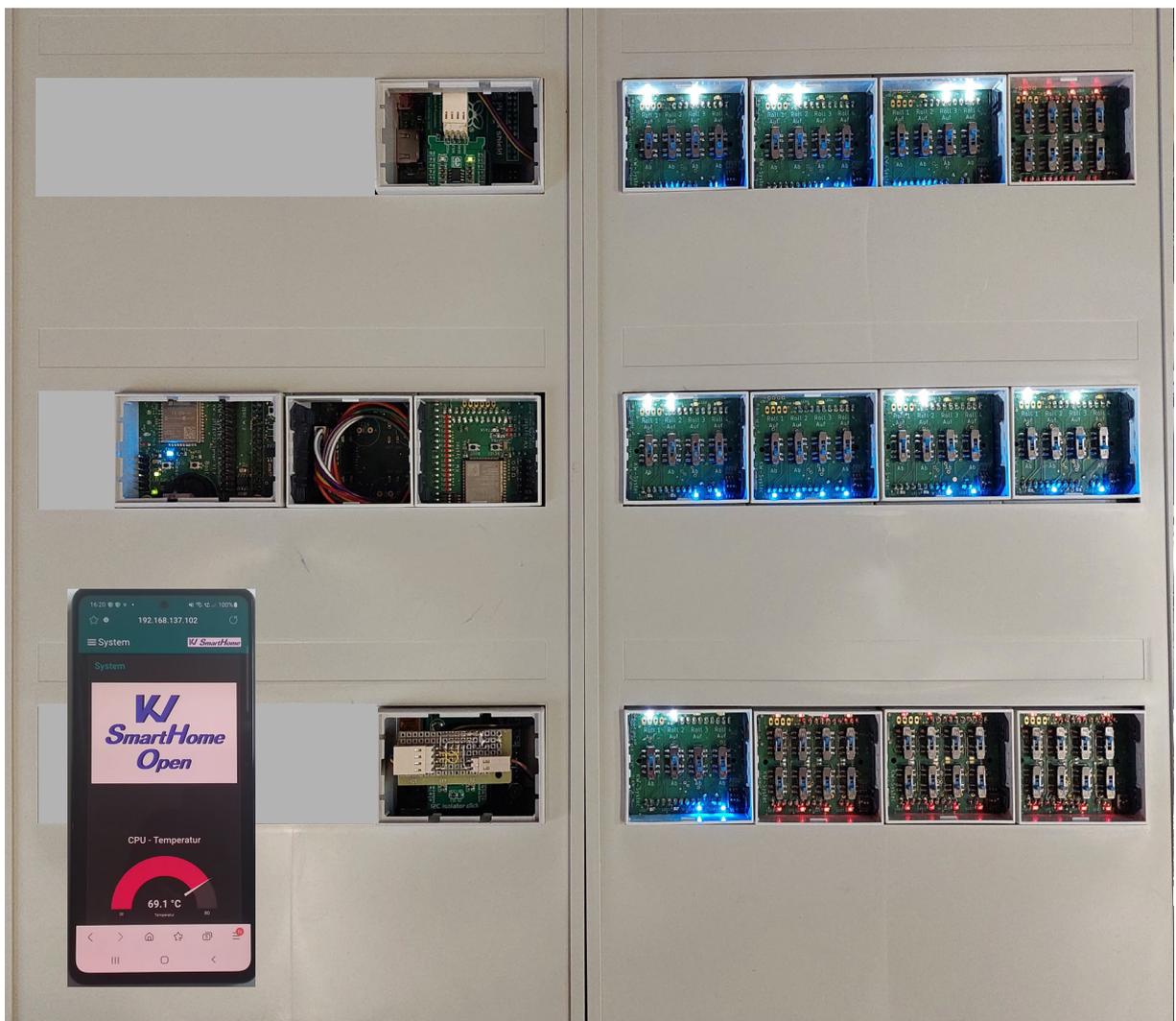


W SmartHome Open





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

Inhalt

KW - SmartHome Open	1
KW-SmartHome Realisierung	3
1.0 Projekt Ziele	5
1.1 Hardware	5
1.2 Software	5
1.3 Dokumentation	6
2.0 Projekt Voraussetzungen	7
2.1 Gebäude Struktur und Elektroinstallation	7
2.2 TCP/IP - Netzwerk Einrichtungen	10
3.0 Software Entwurf	10
3.1 Betriebssysteme	12
3.2 Node-RED Server Software	12
3.3 Zustand Beschreibung von Sensor/Aktor-Geräten	14
3.3.1 Bindungen	15
3.4 Strukturierung der KW-SmartHome Software	17
3.4.1 Programmier- und Bedienoberflächen	22
4.0 Dokumentation	23
2PI4/ESP32 - Server	27
1.0 Hardware Beschreibung	29
2PI_Zero/ESP32 - Server	33
1.0 Hardware Beschreibung	35
Schalter Aktor/Sensor	39
1.0 Hardware Beschreibung	41
2.0 Software Beschreibung	42
3.0 Anschluss Belegung	45
Roll-Laden Aktor/Sensor	47
1.0 Hardware Beschreibung	49
2.0 Software Beschreibung	50
3.0 Anschluss Belegung	53

Dimmer/Motion - Controller	55
1.0 Hardware Beschreibung	57
1.1 PWM-Leistungstreiber Modul PWMC-8	58
1.2 Triac-Leistungstreiber Modul TRIACC-8	61
2.0 Software Beschreibung	62
3.0 Installation.....	65
Relais Modul Schließer	67
1.0 Hardware Beschreibung	69
Relais Modul Wechsler	73
1.0 Hardware Beschreibung	75
Relais Modul Roll-Laden.....	79
1.0 Hardware Beschreibung	81
KW-SmartHome Referenzprojekt „ Wolfer - 13 “	85



KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik
Wolferstraße 13
D - 75031 Eppingen

<http://www.kwsystemtec.de>

KW - SmartHome Open

Karlheinz Wolfmüller

Das Projekt **KW-SmartHome Open** richtet sich an Architekten, Bauherren, Handwerker, Anwender, Planer und System Entwickler, die bei der Gebäude Installation und Automatisierung nicht abhängig sein wollen von Firmen Standards und geschlossenen firmenspezifischen Lösungen mit zu hohen Kosten.

Der SmartHome Bereich zeichnet sich in der Vergangenheit eher dadurch aus, dass etablierten Anbieter von Komponenten und Systemen in der Gebäude Installation und Automatisierung versuchen ihre speziellen Standards in abgeschlossenen Systeme am Markt zu etablieren. Aus der Sicht der Industrie Automatisierung ist der SmartHome Bereich eher ein spezieller Sektor in dem Firmen- und Produkt spezifische Eigenheiten und Merkmale gepflegt werden. Es ist festzustellen, dass Projekte mit professionellen Lösungen mit Produkten von Firmen, die im SmartHome Bereich etabliert sind, aus der Sicht von vielen Bauherren, Architekten und Handwerker zu teuer sind. So ist z.B. ein komplett ausgebautes KNX-Bus System eines ganzen Gebäudes aus Kostengründen oft keine realisierbare Lösung. Die Investitionskosten und die zu erwartenden Betriebskosten, wenn ein System gewartet und gepflegt werden soll, übersteigen meistens die verfügbaren Budgets. Nicht selten sind überzogene Lizenz Bedingungen wesentlich Gründe dafür. An dieser Problematik ist z.B. schon Echelon mit dem LON-Bus System, als technologischer Vorreiter des KNX-Bus Systems in der Europa und Deutschland gescheitert.

In der Technik Geschichte ist häufig zu beobachten, dass hervorragende Technologien an den betriebswirtschaftlichen Rahmenbedingungen und Strategien der Markteinführung scheitern. Die Markteinführung von SmartHome Produkten auf breiter Basis war in der Vergangenheit eher noch keine Erfolgsgeschichte, wenn man das mögliche Marktpotential in Betracht zieht.

An dieser Stelle sollte man noch einen Blick auf die weltweit aktiven Elektronik Bastler und den Low Cost Produktmarkt werfen. Es gibt Angebote, die mit billig Hardware Produkten Funktionalitäten realisieren, die im SmartHome Bereich sehr nützlich sein können. Es gibt Angebote von Sensor- und Aktor- Komponenten, die funktionell sehr interessant sind. Die Tauglichkeit für den Einsatz in einer den Vorschriften und den Richtlinien entsprechenden Gebäude Installation scheitert nicht selten schon daran, dass z.B. Gehäuse Formen nicht kompatibel mit einzuhaltenden Normen in der E-Installationstechnik sind. Handwerker werden schon aus diesem Grund solche Komponenten nicht einsetzen können.

Die Entwicklungen im Mikroprozessor Bereich in den letzten Jahren war so rasant, dass mittlerweile extrem Kosten günstige Produkte mit außerordentlicher Prozessor Leistung verfügbar sind. Die Raspberry PI Produkt Serie mit den diversen Arm-Prozessoren oder die Espressif ESP-Prozessor Familie sind dafür exemplarische Beispiele. Durch gezielte Maßnahmen bei der Hardware Integration können mit diesen Produkten Industrie taugliche Lösungen entwickelt werden. Es lassen sich

dann damit sehr komplexe und aufwendige Software Konzepte realisieren, die auch im SmartHome Bereich außerordentliche Chancen für Kosten günstige Lösungen bieten.

KW SmartHome Open

KW-SmartHome Realisierung



Raspberry Pi OS (64-bit)
A port of Debian Bullseye
Veröffentlicht: 2022-04-01



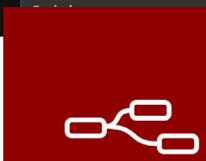
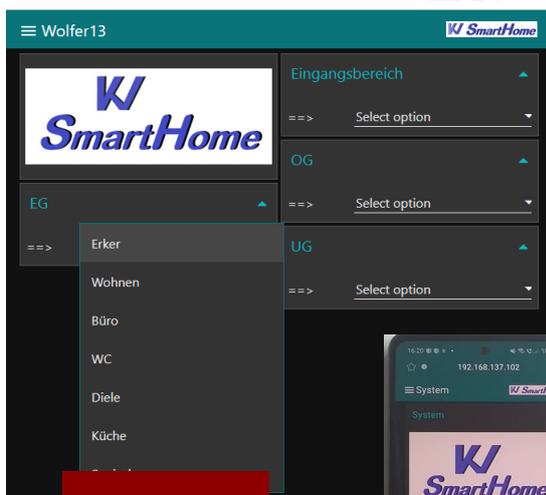
ESPRESSIF



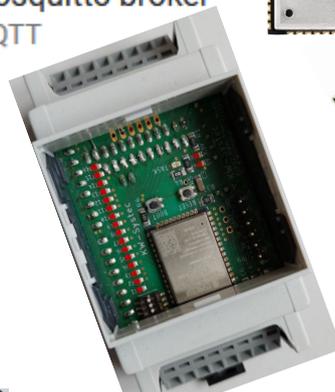
mosquitto



Mosquitto broker
MQTT



Node-RED





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

KW-SmartHome Realisierung

In diesem Teil der Dokumentation wird die Umsetzung des KW-SmartHome Open Projekts beschrieben. Ein wesentlicher Teil dabei befaßt sich mit der Beschreibung der Realisierung des Software Konzepts. In den darauf folgenden Teilen dieser Dokumentation werden die entwickelten Hardware Komponenten des Projekts beschrieben.

1.0 Projekt Ziele

1.1 Hardware

Die im Projekt verwendeten Hardware Komponenten wurden entwickelt mit dem Ziel, optimale Rahmenbedingungen für die Realisierung der Basisfunktionen eines SmartHome Projekts bei minimalen Hardware Kosten zu gewährleisten. Es wird besonders Wert darauf gelegt eine maximale Zuverlässigkeit des Systems zu gewährleisten. Es werden spezielle

1. **Redundanz** Prinzipien

und die

2. Richtlinien der **galvanische Entkopplung** in der industriellen Automatisierungstechnik

umgesetzt. Damit werden optimale Voraussetzung in der Gebäude Installation geschaffen, die Empfindlichkeit gegenüber EMV - Störungen zu minimieren. Dadurch wird eine maximale Betriebssicherheit gewährleistet.

1.2 Software

Die Software Realisierung des Projekts erfolgt ausschließlich mit **Open Source** Standard Tools der **IIoT** - Automatisierung (**I**ndustrial **I**nternet **of** **T**hings). In der Industrie Automatisierung ist eindeutig ein Trend zur Realisierung von professionellen **Open Source Software** Lösungen festzustellen. Damit wird eine ständige Weiterentwicklung der verfügbaren Werkzeuge und realisierter Lösungen vorangetrieben. Weltweit wächst die **Open Source Benutzergruppe** ständig. Viele Firmen folgen mit der Realisierung von Software Projekten diesem Trend.

Im KW-SmartHome Projekt werden grundsätzlich keine Tools und Komponenten eingesetzt, die Lizenz Kosten verursachen. Im Kapitel „Software Entwurf“ auf Seite 10 in der Abbildung 5 auf Seite 11 sind wesentliche Tools zusammengestellt, die diese Bedingungen für das Projekt erfüllen. Alle eingesetzten Tools sind Open Source Werkzeuge und zur allgemeinen Nutzung freigegeben. Damit werden Hersteller spezifische Abhängigkeiten vermieden. Dieser Grundsatz gehört zu den Prinzipien des KW-SmartHome Projekts. Die Voraussetzungen und die Performance dieser

Werkzeuge reicht aus, um alle notwendigen Funktionen bei geforderter Qualität umzusetzen.

Grundsätzlich existieren keine Einschränkungen bzgl. der Auswahl der in einem Projekt verwendeten Hardware Geräte Komponenten. KW-SmartHome Projekte sind grundsätzlich Hardware offen. Neben der Gewährleistung der Funktionalität ist ein Node-RED Interface die Voraussetzung für eine Integration. Schnittstellen für fast alle Markt üblichen Systeme und Komponenten sind z.B. über die Open Source SmartHome Management Systeme **HomeAssistant** oder **IOBroker** vorhanden. Die Leistungsfähigkeit z.B. des 2PI/ESP32-Servers ist ausreichend um das **HomeAssistant** oder **IOBroker** Server System in einem KW-SmartHome Projekt zu integrieren. In **Github** steht System- und Applikations-Software für LINUX-Systeme zur freien Nutzung zur Verfügung. Neben der **Node-RED** Software Entwicklungsumgebung kann **Visual Studio Code** mit integrierter **Platformio** und das **Arduino Framework** eingesetzt werden. Über die **Arduino Libraries** werden alle gängigen Sensor- und Aktor-Komponenten für die **C-Code** Generierung für **ESP32**-Prozessoren unterstützt. Im KW-SmartHome Projekt werden diese Software Tools auf einem Hostrechner eingesetzt.

1.3 Dokumentation

Projekt begleitend wird Wert auf die Erstellung der Dokumentation des Projekts gelegt. Dazu gibt es die Dokumentationen der

Hardware Komponenten:

- „2PI4/ESP32 - Server“ auf Seite 29
- „2PI_Zero/ESP32 - Server“ auf Seite 35
- „Schalter Aktor/Sensor“ auf Seite 41
- „Roll-Laden Aktor/Sensor“ auf Seite 49
- „Dimmer/Motion - Controller“ auf Seite 57
- „Relais Modul Schließer“ auf Seite 69
- „Relais - Modul Wechsler“ auf Seite 75
- „Relais Modul Roll-Laden“ auf Seite 81

Die **Software** Dokumentation

ist implizit durch die **Node-RED Flow-Baumstruktur** mit den **Flow Quellcode** Seiten gegeben. Darin sind die Bezeichnungen der verwendeten

- **Geräte** Komponenten
- **Aktor-** und **Sensor Bezeichnungen** und die
- **Bindungen** der Sensoren und Aktoren

direkt in den Quellcode Flows gemäß der Definitionen im **Gebäude Namensraum** zu entnehmen. Die Flow-Struktur mit den **Nodes** und den **logischen Verbindungslinien** sind selbst erklärend.

Im Referenz Projekt

- „KW-SmartHome Referenzprojekt „ Wolfer - 13 ““ auf Seite 87

können exemplarisch anhand der Beispiel Flows einzelnen Teile der Dokumentation des Projekts eingesehen werden.

2.0 Projekt Voraussetzungen

2.1 Gebäude Struktur und Elektroinstallation

Das Software Konzept und die daraus resultierende Programm Strukturierung richtet sich strikt an der Gebäudestruktur Abbildung 1 auf Seite 7 und Abbildung 2 auf Seite 8 eines SmartHome Projekts aus. Durch das abstrahierte **Gebäude-Modell** mit

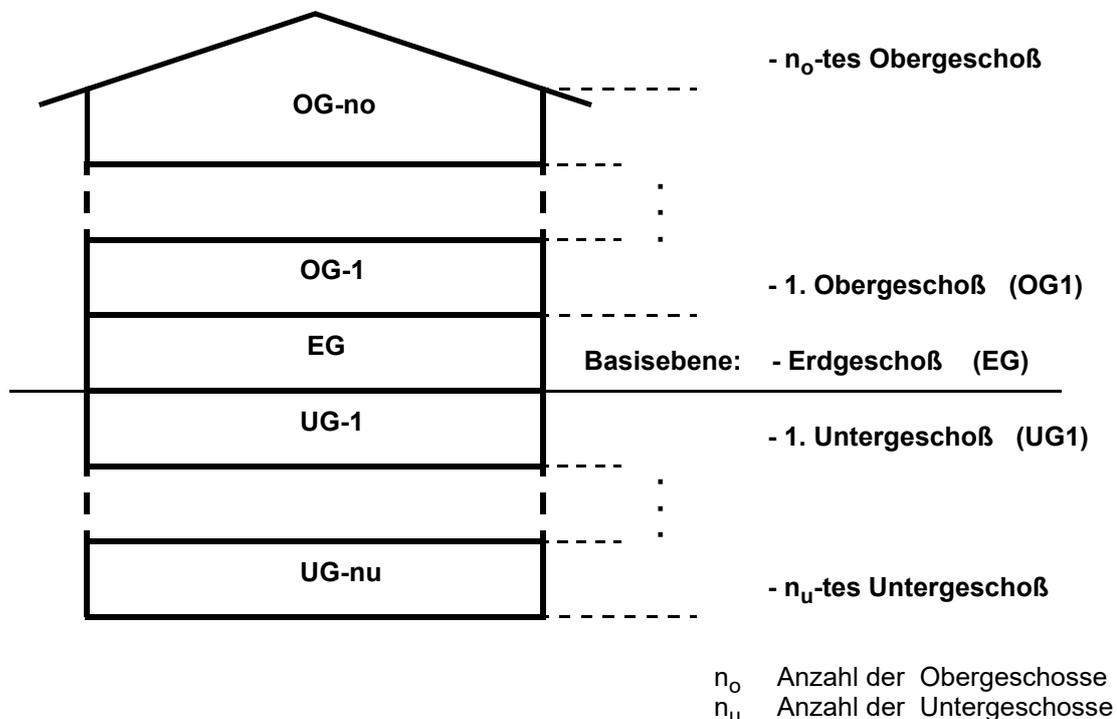


ABBILDUNG 1: Vertikaler Gebäude-Schnitt des abstrahierten Gebäudes- Modell mit festgelegtem Namensraum der Geschosß Bezeichnungen

den definierten Namen der Etagen und der Namen der Räume mit den Raum Objekte in den Grundrissen der Etagen ist der **Namensraum** des gesamten Gebäudes des Projektes gemäß Abbildung 3 auf Seite 9 vollständig definiert. Mit diesem Namensraum ist die Baumstruktur der Applikationen Software für die Programmierung eindeutig vorgegeben und festgelegt.

Die Elektroinstallation basiert auf einem oder mehreren Gebäude- und/oder Etagen-Verteiler. Die optimale Verkabelung erfolgt sternförmig. (s. Abbildung 2 auf Seite 8) Der Verteiler ist der Knotenpunkt an dem alle Leitungen der Leistungsverbraucher des 230V AC-Netzes zusammengeführt sind und direkt oder mit entsprechenden

Schaltelementen (z.B. Relais) mit den Phasenleiter L1, L2 und L3 verbunden sind. Eine optimale Bedienfunktion an den Schaltpunkten der einzelnen Räume wird durch eine sternförmige Installation von Steuerleitungen (z.B. vom Typ CAT 6) erreicht. Der Betrieb kann gemäß den Standards in der Industrie-Automatisierung mit einem +24V/DC Steuerungssystem erfolgen. Marktübliche Schalter und Taster können als

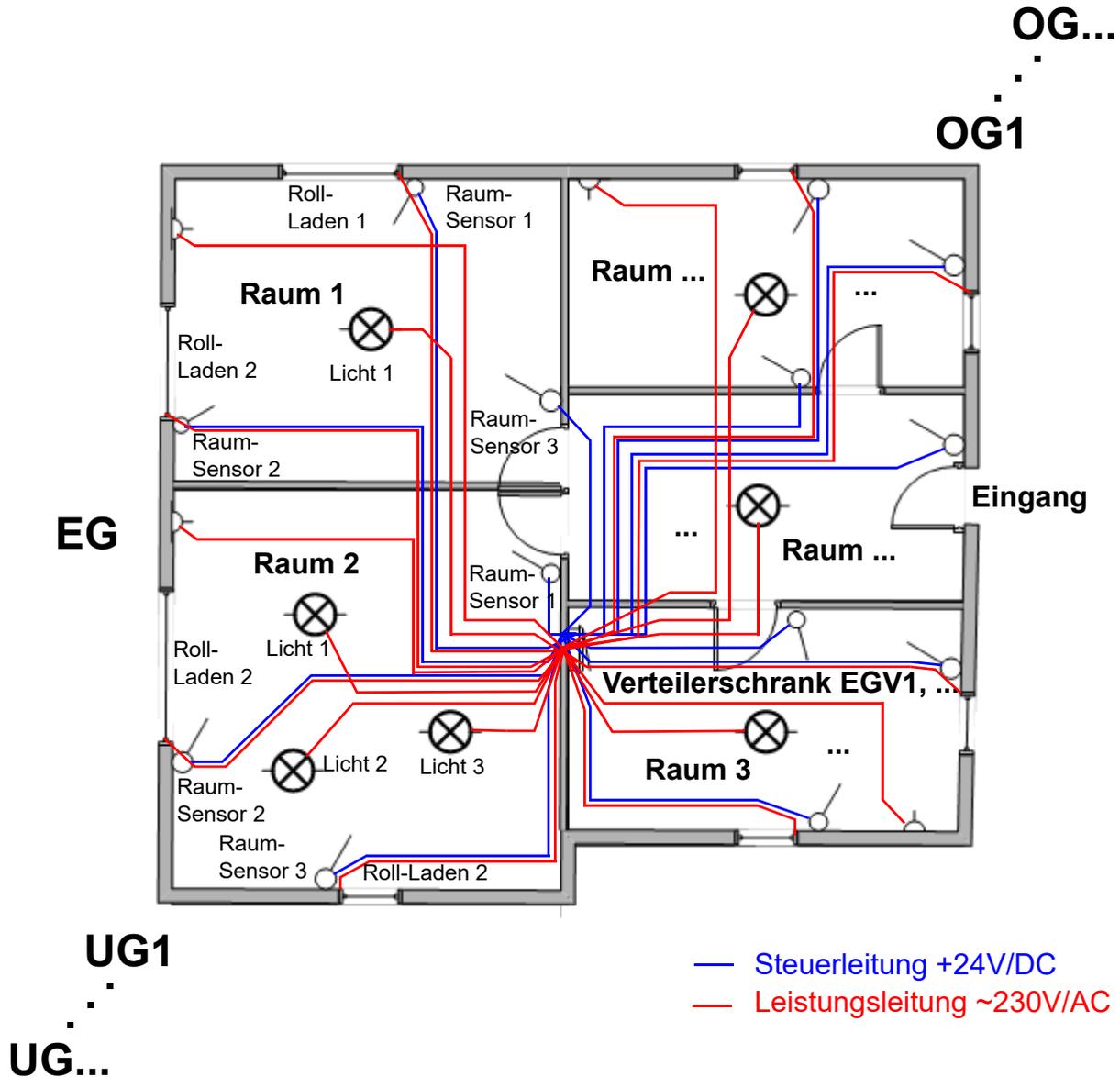
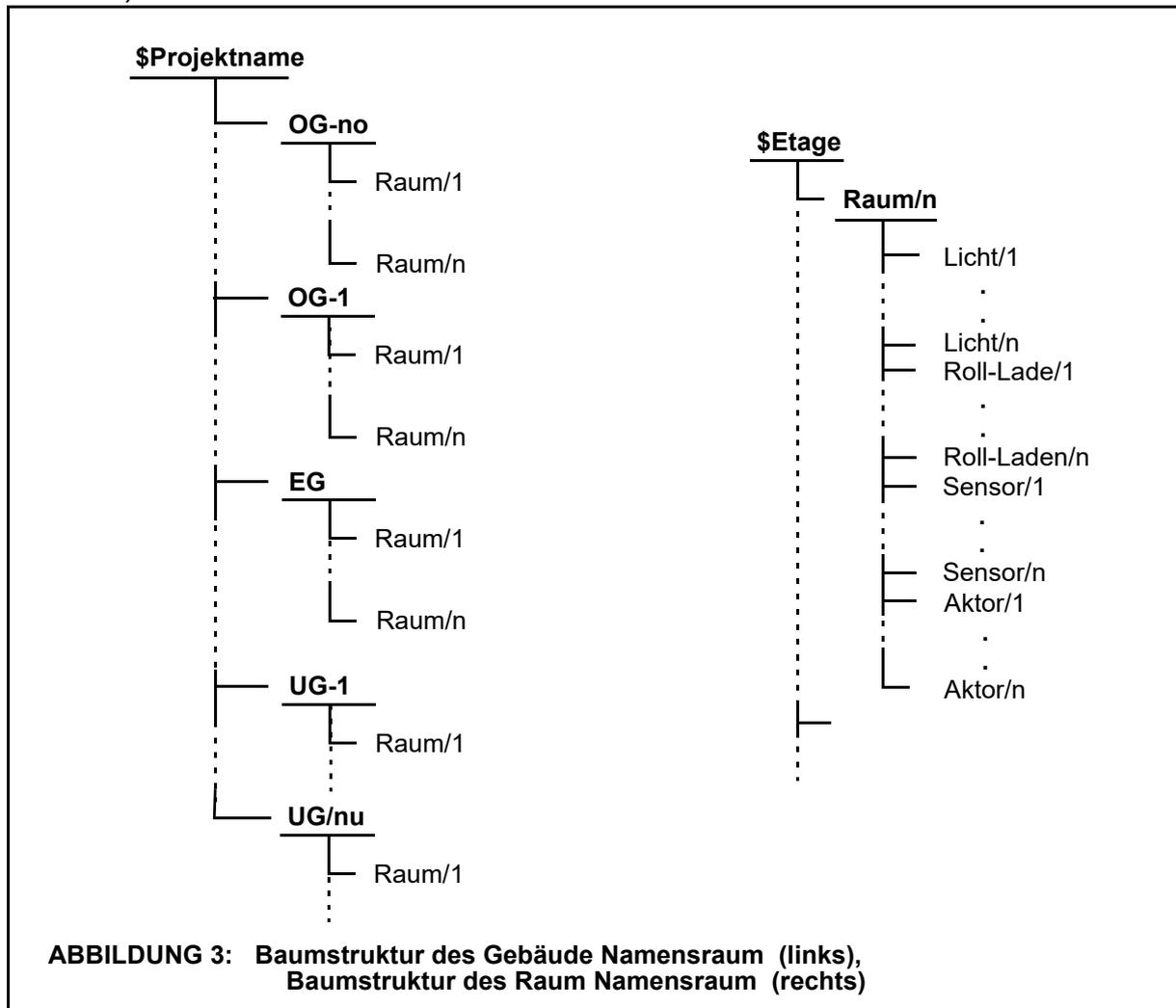


ABBILDUNG 2: Grundrisse der Gebäude Ebenen mit Raumbezeichnungen und Raumobjekten. Stern-Topologie der Elektroinstallation des 230V/AC-Netzes und des +24V-Steuerungssystems (==> Empfohlene Topologie der Leitungsinstallation)

Bedienkomponenten eingesetzt werden. Damit ist bei Beachtung der Installationsrichtlinien die Voraussetzung geschaffen eine sehr zuverlässige und sichere Gebäude Installation zu realisieren. Da mittlerweile auch in der Gebäude Automatisierung zur Leistungssteuerung und Regelung von Verbrauchern zunehmend Frequenz gesteuerte Leistungshalbleiter eingesetzt werden, wird mit dem Konzept ein optimaler Schutz gegen EMV-Störungen erreicht. Der Trend zur leitungslosen Informationsübertragung von Sensor- und Aktorsignalen z.B. über ein WLAN-Netz kann jederzeit parallel dazu betrachtet und betrieben werden. Beim Einsatz des IIoT-Protokolls **MQTT** sind durchaus auch Mischsysteme denkbar. Mit

dem Konzept ist eine maximale Flexibilität im Betrieb und für zukünftige Erweiterungen und Ergänzungen der Funktionalität gewährleistet. In Abbildung 2 auf Seite 8 ist beispielhaft das Konzept der Leitungsinstallation dargestellt. Im Zentralverteiler EGV1 sind damit die optimalen Voraussetzungen für den Einsatz des **2PI/ESP32-Servers** als SmartHome Server und der Stellglied Geräte z.B. für die Relais gegeben. Auf eine strikte Trennung des 230V/AC-Netzes der Leistungsverbraucher und des 24V/DC-Steuerungsnetzes zur Informationsverarbeitung ist zu achten. (**!!! Keine Verbindungen** zwischen **System-GND** der DC-Systeme und **N-Leiter** des 230V/AC-Netzes ==> Bedingung für Netzteil Auswahl)

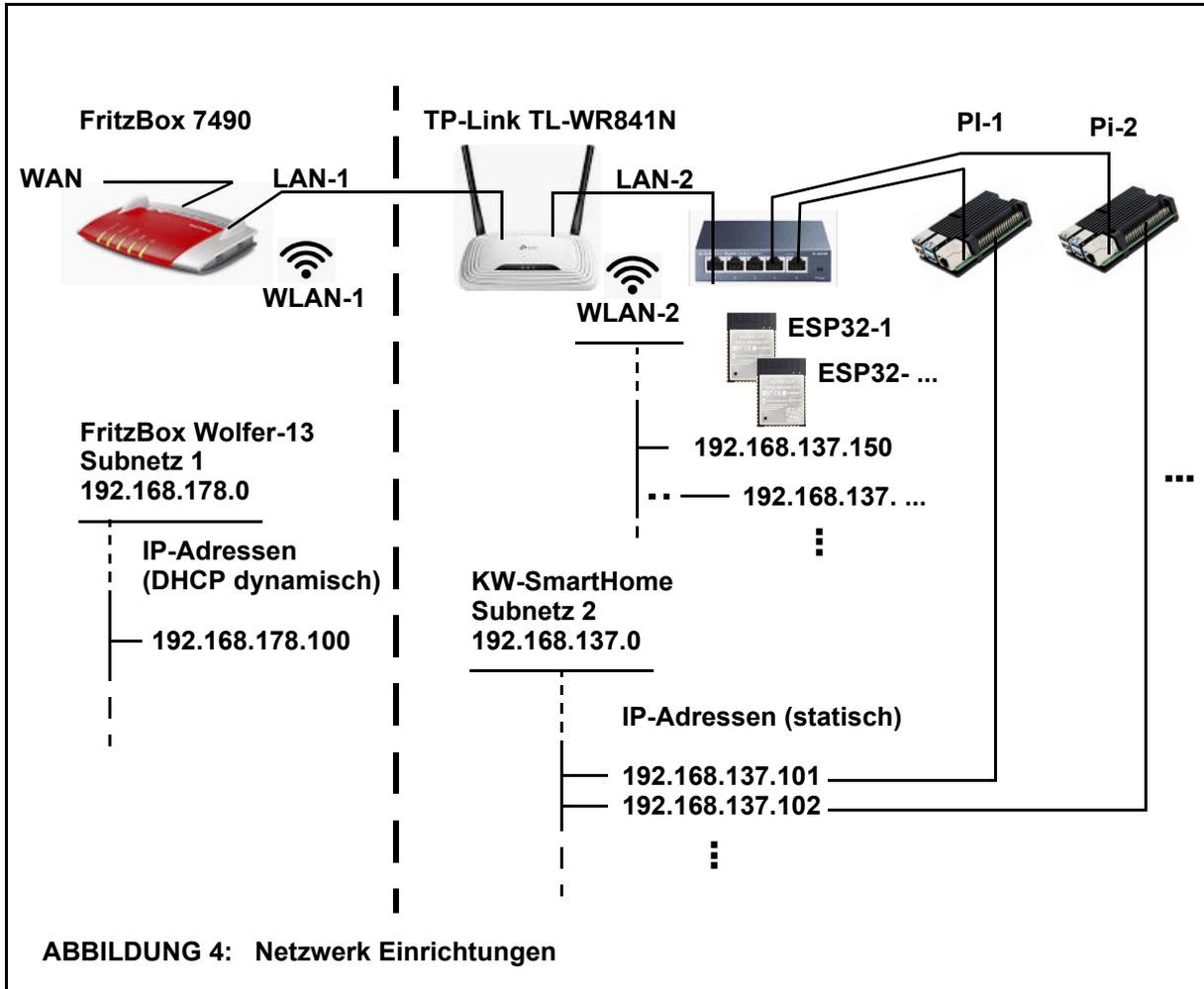


Bei der Verdrahtung der Klemmen der Netzteile der Server Speisung und der Treiber-Geräte erfolgt die Zuordnung der Leistungsverbraucher auf die 3 Phasen des 230V/ AC Netzes. Damit werden die Datenpunkte im Verteiler definiert, die bei der Programmierung relevant sind. An dieser Stelle besteht die Möglichkeit mit dem Verteiler Namensraum den Bezug der Datenpunkt-Variablen in einer logische Verknüpfung mit den physikalischen Geräten vorzunehmen. Mit der **Hardware Verdrahtung** werden direkt auch die notwendigen **Bindungen** in der Software festgelegt. Bei der Programmierung ist eine strenge Trennung von **Geräte-Treiber** Funktionen (Sensoren/Aktoren im Verteiler) und den Sensoren/Aktoren in den Räumen mit den Gebäude Betriebsfunktionen (**Raum Applikationen**) vorzunehmen. Es ergeben sich exakt definierte Schnittstellen bei der Realisierung der

Verbindungslogik zwischen Sensor- und Aktor-Funktionen. Die exakte Beschreibung der Verbindungslogik erfolgt später im Kapitel „Software Entwurf“ auf Seite 10 .

2.2 TCP/IP - Netzwerk Einrichtungen

In Abbildung 4 auf Seite 10 ist das Netzwerk Konzept dargestellt. Das KW-



SmartHome Subnetz ist konsequent getrennt vom Heimnetz des FritzBox 7490 Routers. Sämtliche IP-Netzwerk Adressen, die in den Software Modulen des KW-SmartHome Projekts verwendet werden liegen im Bereich des 192.168.137.0 Subnetzes. Mit einer Unterbrechung der Leitungsverbindung zum LAN-1 Subnetz kann das KW-SmartHome Subnetz auch als vollständig isoliertes lokales Netzwerk bei maximaler Sicherheit betrieben werden. Die IP-Adressen werden statisch festgelegt.

3.0 Software Entwurf

Zum besseren Verständnis der Logik des Software Konzept ist es an dieser Stelle sinnvoll sich mit der Gebäude Struktur im Kapitel „Gebäude Struktur und Elektroinstallation“ auf Seite 7 zu befassen. Durch die Vorgaben des Gebäude Modells ergibt sich automatisch die Software Struktur des konkreten Projekts. Mit den Vorgaben des Gebäude Modells ist der Namensraum des Gebäudes am konkreten Projekt eindeutig festgelegt und damit ist auch die Software Struktur vorgegeben. Die

Erstellung der Software ist eigentlich nur noch ein Vorgang , bei dem systematisch die **Geräte-Treiber** und **Raum-Applikationen** über entsprechende Templates erstellt werden müssen bis der Namensraum des Gebäudes vollständig abgearbeitet ist. Jedes installierte Gerät im Verteiler wird mit einem **Geräte-Treiber Flow** betrieben. Zu jedem Gerät gehört immer ein **Bindungen Flow**, in dem gemäß des Leitungsschaltplans die logischen Bindungen des Geräts mit den Raum Komponenten in der Programmierung vorgenommen werden. Für jeden Raum gibt es ein **Raum Applikationen Flow** in dem die Raum Funktionen implementiert werden. Mit dieser Vorgehensweise ist die Kontrolle der Programmierung der Funktionalitäten leicht nach zu vollziehen und die Vollständigkeit leicht zu prüfen. In Kapitel „Node-RED Server Software“ auf Seite 12 ist die Vorgehensweise ausführlich dokumentiert.

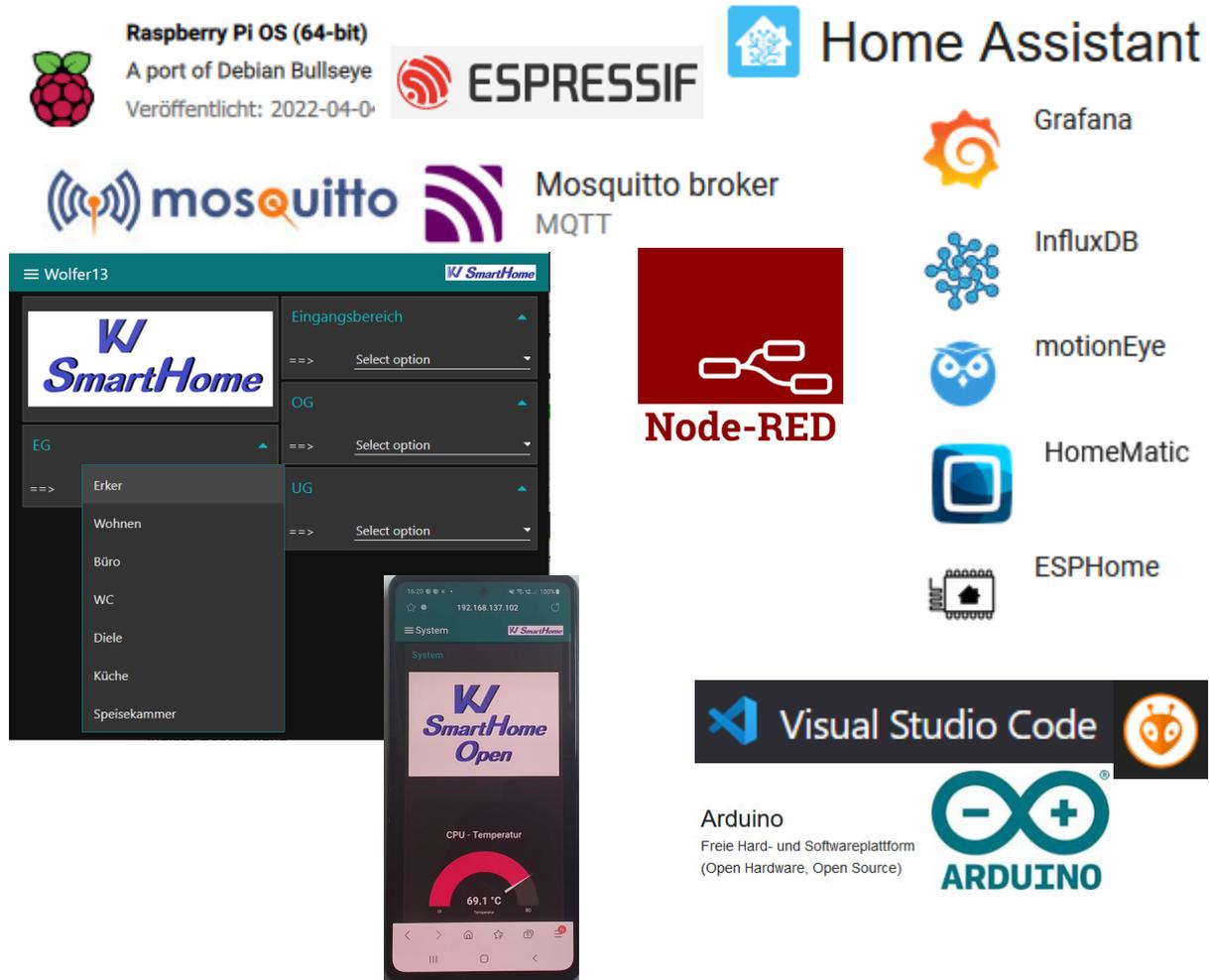


ABBILDUNG 5: Software Tools des KW-SmartHome Projekts

Bei der Erstellung der Software im konkreten Projekt wird automatisch auch die **Dokumentation** erstellt. Das ist eine sehr angenehme Begleiterscheinung der Vorgehensweise. Dieser Vorgang wird erfahrungsgemäß eher vernachlässigt. In dem hier praktizierten Verfahren wird praktisch die Dokumentation durch den Prozess der Programmierung zwangsläufig erstellt.

3.1 Betriebssysteme

Auf den Raspberry PI4 (Siehe „2PI4/ESP32 - Server“ auf Seite 27) ist ein

- Linux Raspian OS(32-bit) System mit Desktop Umgebung

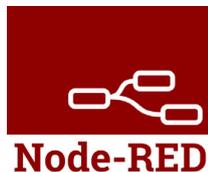
installiert. Aktiviert sind die folgende Server- und Interface Funktionen:

- VNC - Server
- SSH -Server
- I2C - Bus Interface
- SPI - Interface
- Serielles UART-Interface

Es können mit einem VNC-Client von einem Desktop/Laptop-Host eine Browser basierte Netzwerk-Verbindungen mit der entsprechenden Netzwerk-Adressen (s. Abbildung 4 auf Seite 10) zu einem PI4 des 2PI/ESP32-Servers aufgebaut werden. Damit stehen Desktop-Oberflächen zur komfortablen Bedienung der eingesetzten PI-Modulen zur Verfügung.

3.2 Node-RED Server Software

Zur Realisierung der KW-SmartHome System wird ausschließlich das Ereignis gesteuerte **Datenfluss-Tool**



- **Node-RED**

verwendet. Grundlagen Kenntnisse zur Programmierung der **Node-RED Flows** werden hier für die weitere Betrachtung vorausgesetzt. Sollte Bedarf für eine Einführung und eine vertiefte Einarbeitung bestehen kann das anhand der Node-RED Webpage mit dem Link <https://nodered.org> oder diversen Youtube Videos erfolgen.

Die Node_RED Programmierung erfolgt über eine Browser basierte Verbindung



zum Node-RED Server des zu programmierenden Zielsystems mit seiner IP-Adresse und dem Port 1880.

Zur Kommunikation wird das **IIoT** Protokoll

- **MQTT**



eingesetzt. Als MQTT - Broker wird der

- **Mosquitto** - Broker



für Raspberry PI4 verwendet. Diese System Voraussetzungen sind auf allen im KW-SmartHome Projekt beteiligte PI4-Modulen durch eine Systemspiegelung herzustellen. Ergänzend dazu stehen noch zusätzliche Werkzeuge zur Realisierung von Betriebsfunktionen zur Verfügung.

Als Tool zur komfortablen Visualisierung von Sensordaten kann

- **Grafana**



eingesetzt werden.

Zur Archivierung von Sensordaten mit Zeitstempel bietet sich das Echtzeit Datenbanksystem

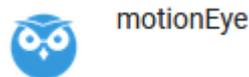
- **Influxdb**



an.

Für das Management von Video Kameras kann

- **MotionEye**



verwendet werden.

Das SmartHome Management System

- **Home Assistant**



als Open Source Tool bietet sich mit seiner Node-RED Schnittstellen Nodes zur Anbindung fast aller SmartHome Komponenten anderer Hersteller und Anbieter an.

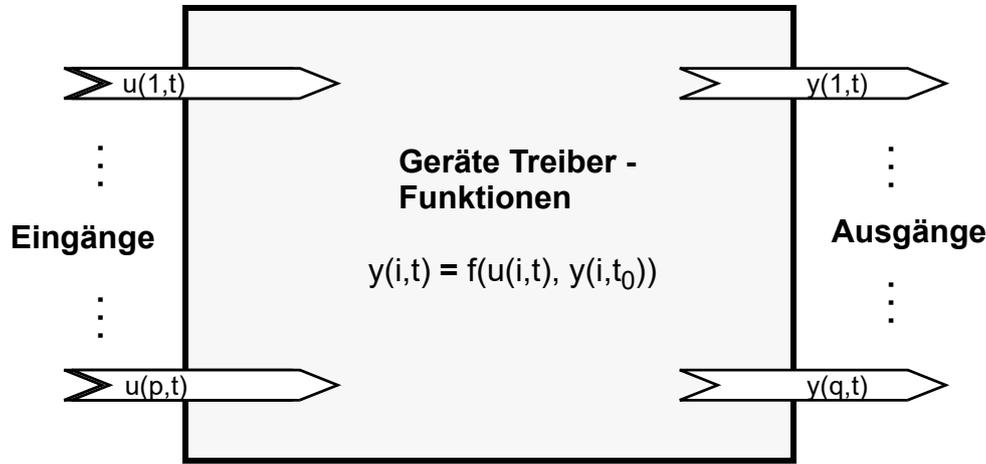
- **IOBroker**



ist ein alternatives frei verfügbares SmartHome Management System, das ebenfalls zur komfortablen, individuellen Gestaltung des eigenen SmartHome Systems eingesetzt werden kann.

3.3 Zustand Beschreibung von Sensor/Aktor-Geräten

Sensor/Aktor Geräte sind physikalische oder virtuelle Komponenten, die ihre Zustände an den **Ausgängen** durch **Geräte Treiberfunktionen** mit den **Eingängen** verknüpfen. Der Zustand am Ausgang $y(i,t)$ ändert sich zu einem Zeitpunkt t mit der



$u(i)$ p Eingangsdatenpunkt Variablen

$y(i)$ q Ausgangsdatenpunkt Variablen

p Anzahl der Eingangsdatenpunkt Variablen
 q Anzahl der Ausgangsdatenpunkt Variablen
 t Zeitvariable

ABBILDUNG 6: Zustandsmodell eines Sensor/Aktor - Geräts

Änderung des Eingangszustands $u(i,t)$ zum Zeitpunkt t gemäß einer Geräte Treiber-Funtion

$$f(u(i,t), y(i,t_0)) .$$

Dabei kann der neu Zustand auch noch vom alten Zustand $y(i,t_0)$ zum vorherigen Änderungszeitpunkt t_0 abhängen. Verwendet man dieses Treiber-Modell in einer Node-RED-, MQTT- Programm Umgebung können die Datenpunkt Variable $u(i)$ und $y(i)$ mit **Topic** Namen Strings adressiert werden. Zum Informationsaustausch mit dem MQTT-Protokoll werden mit der

subscribe(,Topic Namen String')

Funktion die Datenpunkt Variablen beim MQTT-Broker bei der Initialisierung der Geräte Treiberfunktion angemeldet. Zur Laufzeit des Programms können zum Änderungszeitpunkt eines Eingangszustandes $u(i,t)$ die aktuellen Werte der Datenpunkt Variablen $y(i,t)$ an den Ausgängen mit der

publish(,Topic Namen String')

Funktion an die Adresse des ,Topic Namen String' übertragen werden. Es wird dann mit dem aktuellen Wert der Datenpunkt Variablen $u(i,t)$ die Datenpunkt Variable am Ausgang mit der Geräte Treiber-Funktion

$$y(i,t) = f(u(i,t), y(i,t_0))$$

gebildet. Wird nun dieses Geräte Treiber-Modell sowohl für die Zustandsänderungen an einem Sensor als auch bei einem Aktor benutzt und in einem Node-RED Flow umgesetzt lassen sich z.B. Sensor Ausgänge mit Aktor Eingänge über **Bindungen** (Node-RED Verbindungslinien) koppeln. Wenn nun der Aktor Ausgang den aktuellen Aktorzustand mit der Aktor Treiber-Funktion als Aktor Status ausgibt lassen sich z.B. Programmschleifen mit Quittierungsvorgängen realisieren. Bei virtuellen Sensoren

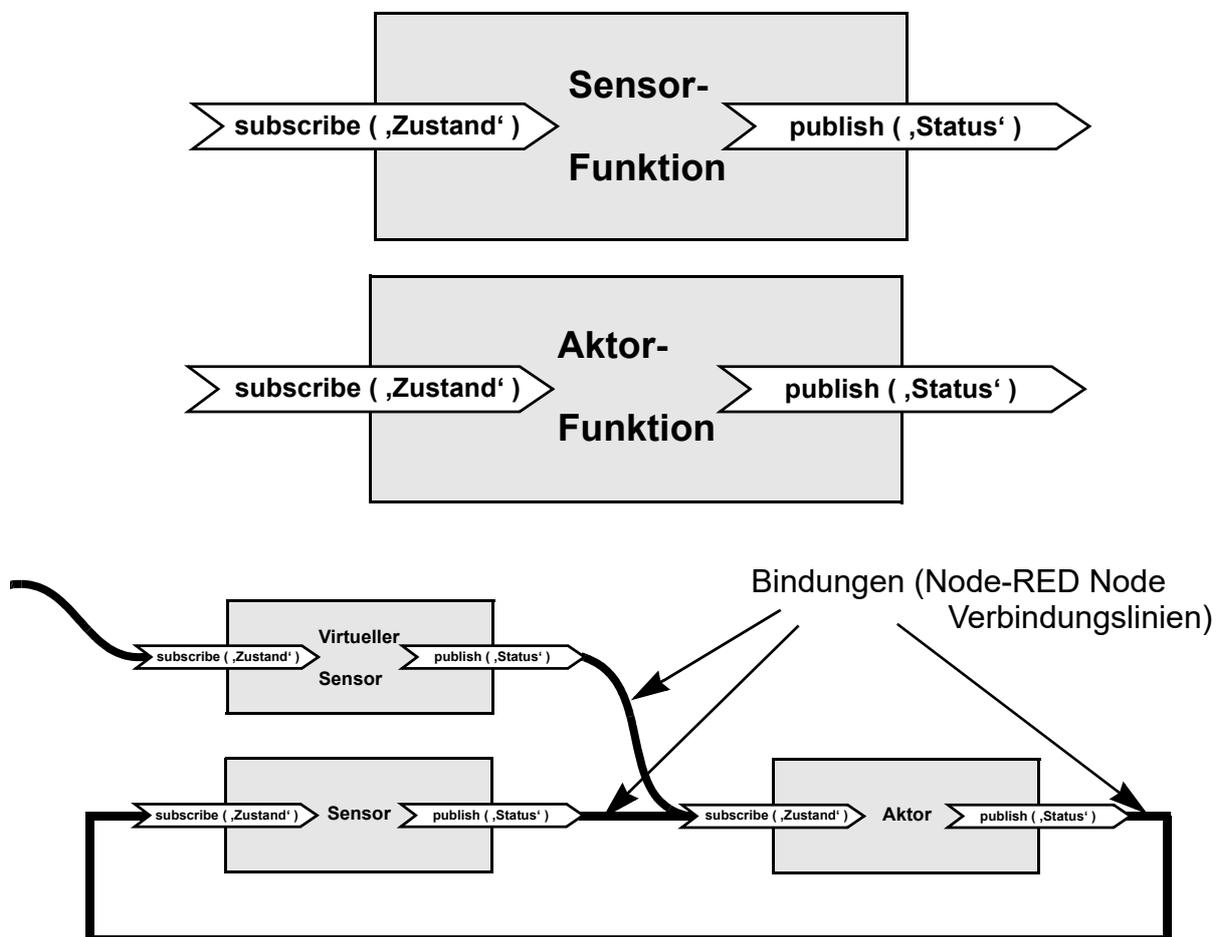
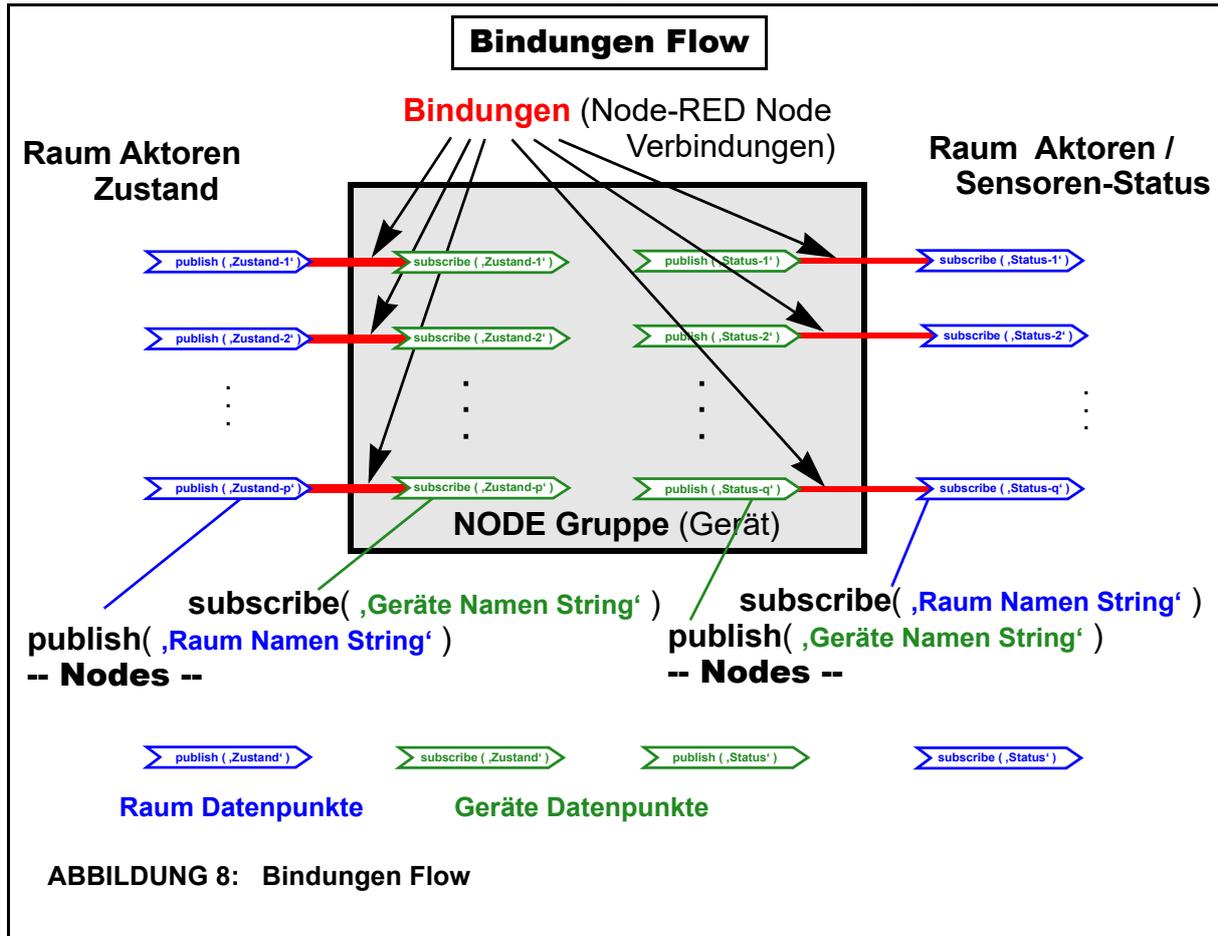


ABBILDUNG 7: Beispiel der Realisierung der Bindungen von Sensoren mit einem Aktor über eine Sensor/Aktor-Kopplung mit z.B. einer Aktor-Status Rückkopplung

lassen sich damit z.B. grafische Objekt Attribute wie Farbumschläge bei Statusänderungen an die realen Aktor Ausgangszustände koppeln. Werden Mehrfachbindungen eines Aktoreingangs mit mehreren Sensorausgängen realisiert wird sich das Aktorverhalten gemäß einer logischen „ODER“-Verknüpfung der Eingangsbindungen ergeben.

3.3.1 Bindungen

Eine zentrale Aufgabe bei der Programm Entwicklung ist die Kopplung der Raum Sensor/Aktor Komponenten und von virtuellen Sensoren mit den installierten Geräten in den Verteiler gemäß Abbildung 2 auf Seite 8. Dabei müssen die Kopplungen in den Datenflüssen exakt der Schaltschrank Instrumentierung und dem Leitungsplan der



werden die Datenfluss Ketten mit den roten Verbindungslinien in einem Node-RED Flow geschlossen. Mit dieser Programmier Logik werden die Datenpunkt Variablen der Geräte (grün) mit den entsprechenden Raum Datenpunkt Variablen (blau) synchronisiert.

Es ist hier besonders darauf hinzuweisen, dass die roten Node-RED Programm Verbindungslinien exakt den physikalische Leitungsverbindungen des Verdrahtungsplans entsprechen. Dieser Sachverhalt ist besonders vorteilhaft bei der Projekt Dokumentation. Die Dokumentationen der Software Entwickler und der Handwerker für die Elektroinstallation sind identisch. Es können alle Bindungen des Gesamtprojekts in den **Bindungen Flows** der Geräte vorgenommen werden. Dazu müssen die Geräte Bindungen Templates bei der Programm Entwicklung in ein Bindungen Flow für jedes Gerät kopiert werden und die Topic Namen Strings entsprechend der Festlegung im Gebäude- und Raum Namensraum angepasst werden. Mit diesem Software Konzept ist zwar der Laufzeit Aufwand des Programms nicht mehr minimal. Die Erfahrung hat aber bisher gezeigt, dass bei den verfügbaren Hardware Voraussetzungen keine Laufzeit Probleme zu erwarten sind. Die Vorteile der strukturellen Identität der Hardware Installation und der Software Struktur mit den einheitlichen Bezeichnungen rechtfertigen den Mehraufwand im Programm Code. Die saubere Funktionen Trennung von Geräte- und Applikationen Software Module erleichtert extrem die Debug Vorgänge bei der Code Generierung. Die erzielbare Transparenz und Vereinheitlichung der Dokumentation von Hardware und Software ist ein weiterer extremer Vorteil des Konzepts.

3.4 Strukturierung der KW-SmartHome Software

In einem KW-SmartHome System sind in einem Gebäude mindestens 1 oder aus Gründen der Redundanz für mehr Sicherheit mehrere MQTT-Broker aktiv (Abbildung 9 auf Seite 17)

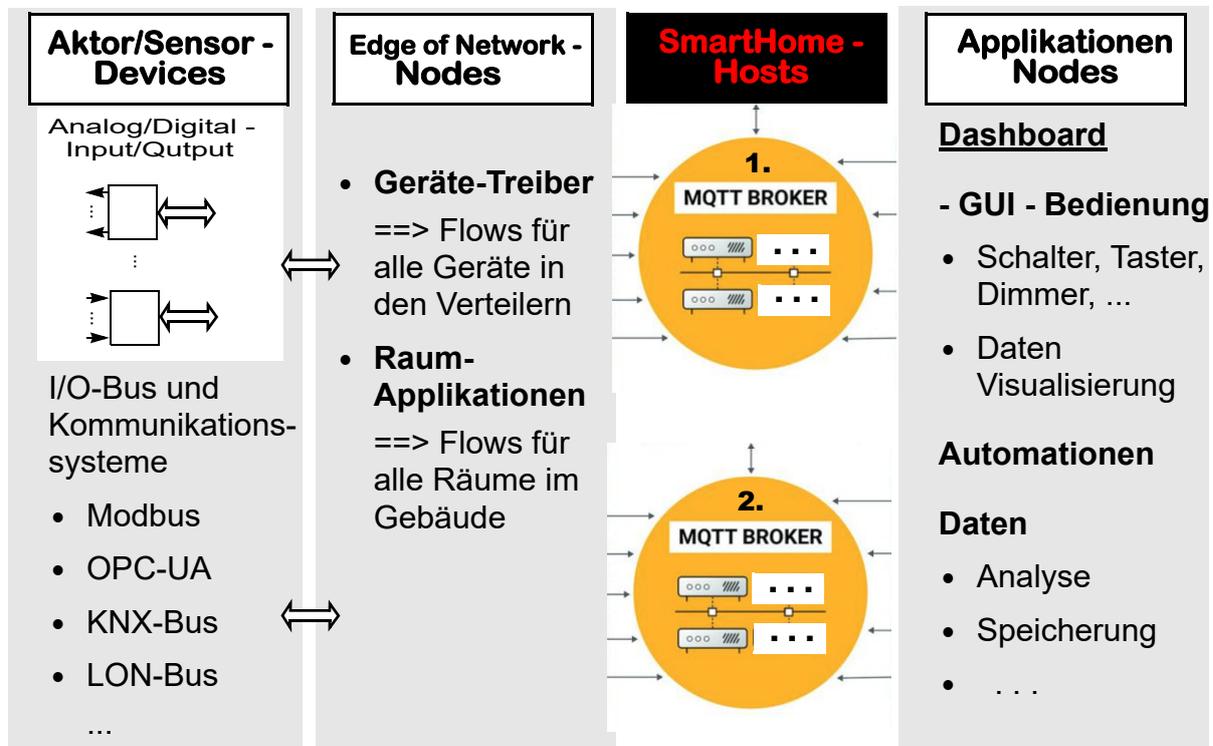


ABBILDUNG 9: MQTT - Broker 1, 2 ==> KW-SmartHome Architektur, redundante Instanzen des MQTT-Brokers im lokalen Subnetz

In jedem Gebäude Verteiler läuft ein Node_RED Server. Der Node-RED Programmcode wird strikt unterteilt in

- **Geräte Treiber Flows** mit **Bindungen Flows**,
- **Raum Applikationen Flows** und
- **Allgemeine System Flows**.

Die **Geräte Treiber Flows** sind für die Aktivierung der im Verteiler installierten Geräte und zum Betrieb der im Gerät zur Verfügung stehenden Aktor-Ausgänge (==> Relais Kontakte) und Sensor-Eingängen (Schalter/Taster/Dimmer/ ...) zuständig. Jedes Gerät hat eine im **I2C/SPI- Bus** System festgelegte Geräte Adresse. Mit dieser Geräte Adresse, dem Namen des Verteilers und einer Aktor Nummer des Geräts kann eine **Datenpunkt Variable** eines bestimmten Typs durch einen **Topic Namen** referenziert werden, mit der ein Zugriff auf den Aktor/Sensor-Zustand möglich ist. Im Namensraum des Gebäudes ist jeder Aktor und Sensor eindeutig identifiziert. Im Geräte Treibercode wird jeder Aktor mit seinem Topic Namen mit einer MQTT **Subscribe-Funktion** beim MQTT-Broker angemeldet. Damit ist jeder Aktor über das MQTT-Protokoll erreichbar. Der Zustand jedes Aktors und Sensors wird bei jeder Änderung mit einer MQTT **Publish-Funktion** auf eine entsprechenden Datenpunkt Variablen

eines bestimmten Typs aktualisiert. In Abbildung 10 auf Seite 18 ist ein Beispiel eines Geräte Flows dargestellt. Für jeden Roll-Laden ist ein Zustands-Automat programmiert, der nur 3 Zustände exklusiv annimmt:

3. „Stopp“ ==> kein Kontakt aktiv,
4. „Auf“ ==> Auf-Kontakt aktiv,
5. „Ab“ ==> Ab-Kontakt aktiv .

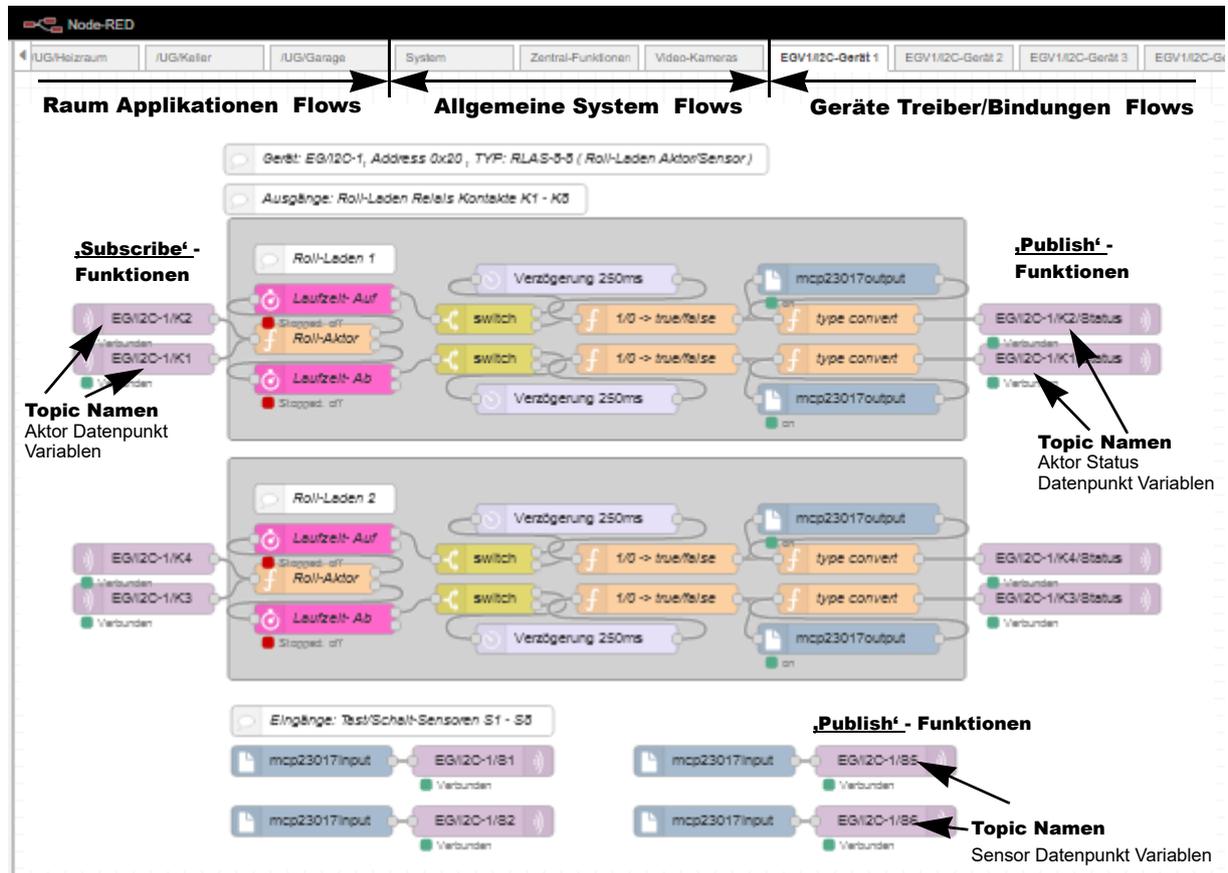


ABBILDUNG 10: Geräte Treiber Flow: Beispiel Rolladen Aktor/Sensor für Gerät RLAS-8-8

Die Fahrzeit ist für jeden Roll-Laden getrennt einstellbar. Durch Hardware- und Software Verriegelung ist ausgeschlossen, dass zu einem Zeitpunkt der „Auf“- und „Ab“- Aktor Kontakt gleichzeitig aktiv ist. Die Zustandsänderungen werden durch Änderung der Sensor Zustände an den Eingänge veranlaßt. An den Eingängen des Geräts (Abbildung unten) werden die Sensor Zustände (physische Schalter/Taster) eingelesen.

In den **Bindungen Flows** (Beispiel Abbildung 11 auf Seite 19) werden die **Bindungen der Raum Sensoren/Aktoren** mit den **Geräte Sensoren/Aktoren** über ihre **MQTT-Topic Namen Strings** vorgenommen. Damit können die Flow-sheets der Bindungen Flows als Bindungen Listen in der Dokumentation direkt verwendet werden. Die Dokumentation erstellt sich also automatisch mit der Programmierung. Wenn der Funktionstest des gesamten Programmsystems erfolgreich durchgeführt werden konnte, ist folglich die Dokumentation immer Fehler frei. Die Qualität der

Dokumentation ist nicht mehr von der Disziplin der Software Entwickler und/oder der zuständigen Handwerker für die Instrumentierung und Installation abhängig.

Die **Raum Applikationen Flows** werden zur Realisierung der Raumfunktionen erstellt. Jeder Raum des Gebäudes ist im Namensraum eindeutig identifiziert. Für jeden Raum wird genau ein Raum-Flow mit dem Flow-Namen des Raum-Namens im Namensraum des Gebäudes erstellt. Im Code des Flows wird jede im Raum

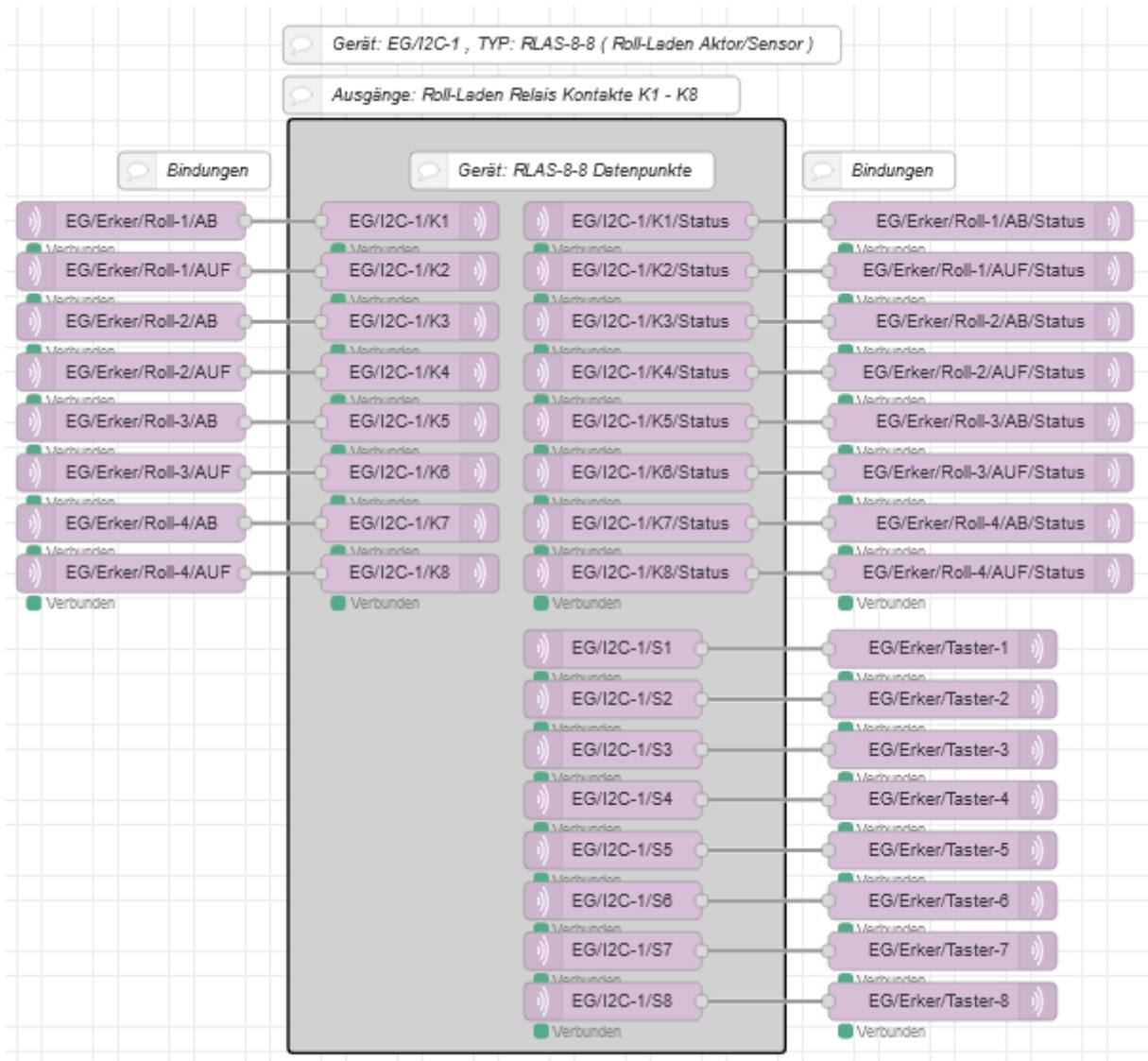


ABBILDUNG 11: Beispiel Bindungen Flow zum Roll-Laden Geräte-Treiber Flow (Abbildung 10 auf Seite 18)

installierte Sensor- und Aktorfunktion mit den verfügbaren Node-RED nodes und den entsprechenden Verbindungen zur Verfügung gestellt. In der linearen Baumstruktur der Node-RED Menü Zeile wird die Flow Reihenfolge sinnvoller Weise nach dem Top-Down-Prinzip der Raumposition im Gebäude Namensraum für die Etagen von oben nach unten angeordnet. Als erster Flow (s.Abbildung 12 auf Seite 20) wird immer ein Flow mit dem Namen des Projekts codiert, in dem die Raum Reihenfolge über eine

Dashboard Menü-Funktion festgelegt wird. Die Dashboard -Funktionen zur Anzeige

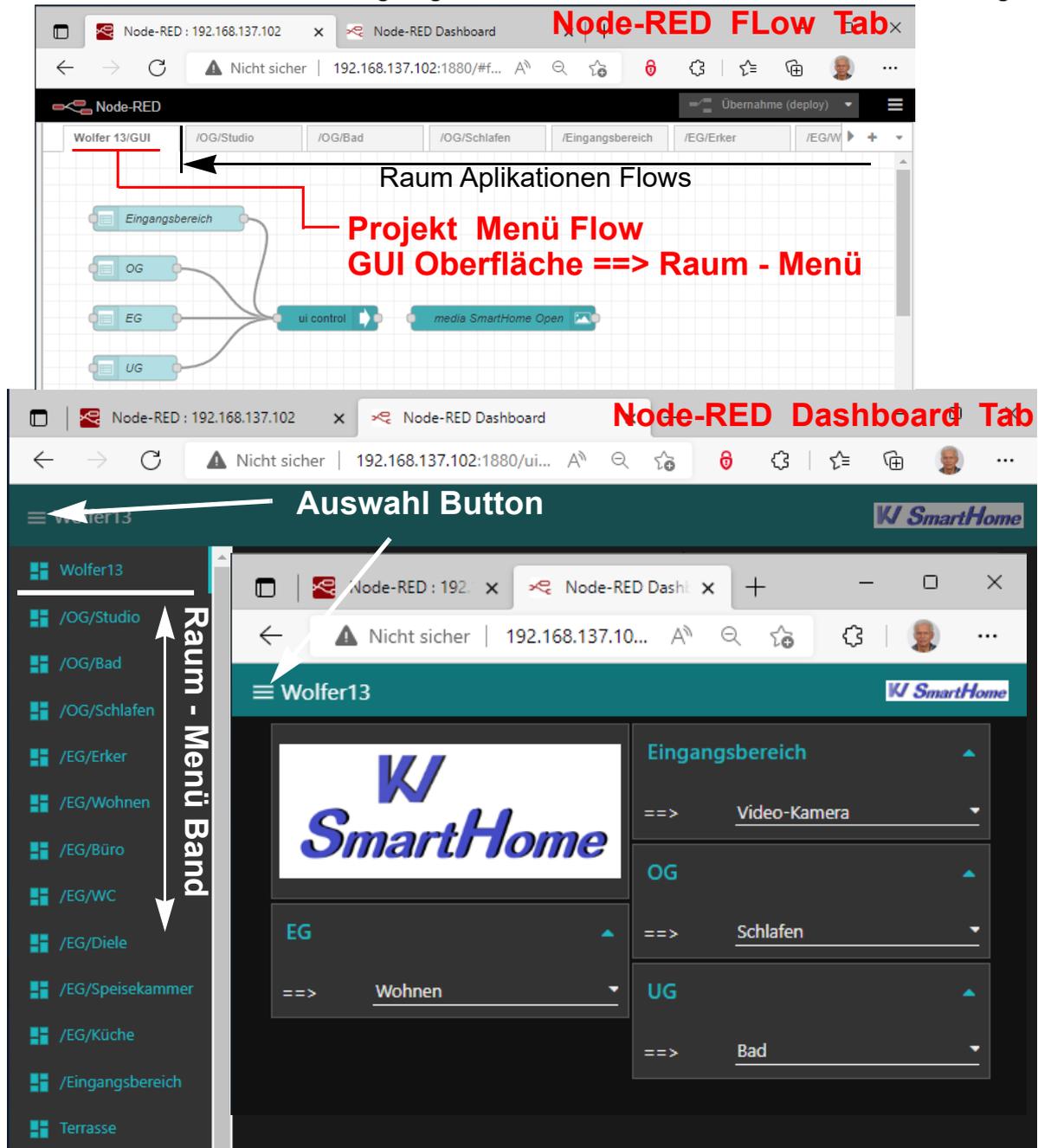


ABBILDUNG 12: Node-RED Flow Tab ==> Menüzeile (oben);
 Node-RED Dashboard Tab ==> GUI - Anzeige (unten):
 Projekt-Button
 ==> Raum Auswahl Menü
 ==> Raum Menü Band

der grafischen Raumobjekte werden ebenfalls in jedem Raum Flow für den entsprechenden Raum programmiert. Die grafisch Bedienoberfläche ist so organisiert, dass in der Anzeige durch Aktivierung des **Auswahl Buttons** am linken Rand der Dashboard Menü Zeile sich ein scrollbares Auswahl Menü Band aller verfügbaren Räume in den einzelnen Etagen aufbaut, in dem ein gewünschter Raum

mit seinen Raum-Funktionen ausgewählt werden kann. In Abbildung 13 auf Seite 21

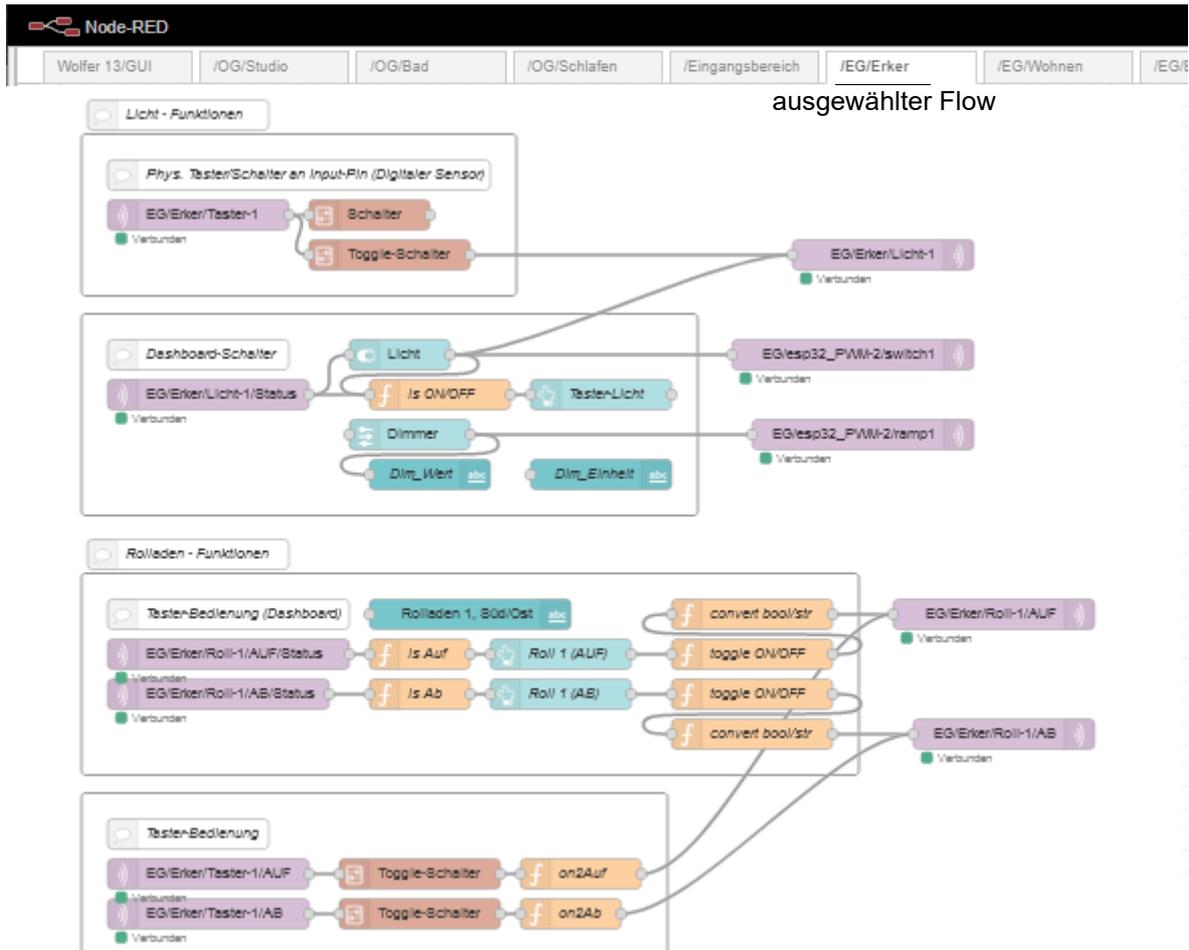


ABBILDUNG 13: Raum Applikation Beispiel: - Lichtfunktion Schalter/Dimmer - Roll-Laden Funktion

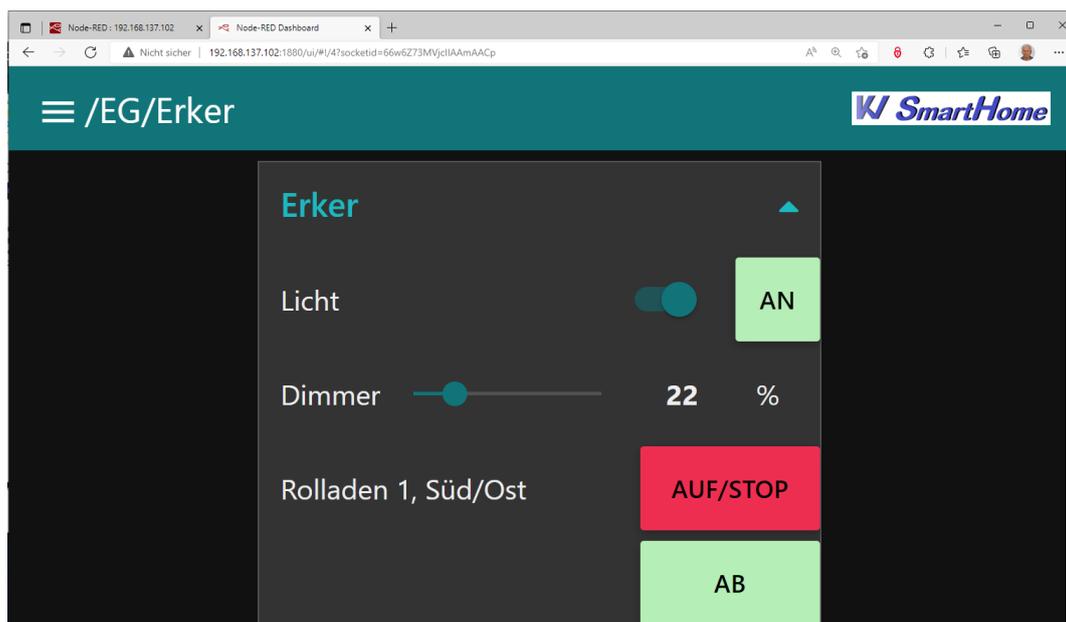


ABBILDUNG 14: Dashboard Anzeige von Flow Abbildung 13 auf Seite 21

ist ein Teil z.B. der Raum Applikation /EG/Erker mit der resultierenden **Dashboard Grafik** Abbildung 14 auf Seite 21 dargestellt. Jeder physische Schalter und Taster Sensor im Raum wird zusätzlich noch mit einem jeweiligen **virtuellen Sensor** als Dashboard Objekt aus der Node-RED Dashboard Node Library ergänzt. Damit stehen in Endgeräten mit **grafischen Bedienoberflächen** virtuelle **grafische Sensor Objekte** zur Zustandsänderung von Aktoren zur Verfügung. Physische Schalter/ Taster und Dimmer Einheiten sind damit bei der Bedienung über ein visuelles Endgerät nicht notwendig. Die Darstellung des Zustandes mit virtuellen grafische Objekten kann vielfältiger, komfortabler und nach persönlichen Wünschen individuell gestaltet werden.

In den **Allgemeine System Flows** werden z.B. Schaltfunktionen für Sonnen Auf- und Untergänge oder Zentrale Licht- und Roll-Laden Funktionen realisiert. Ein Beispiel ist

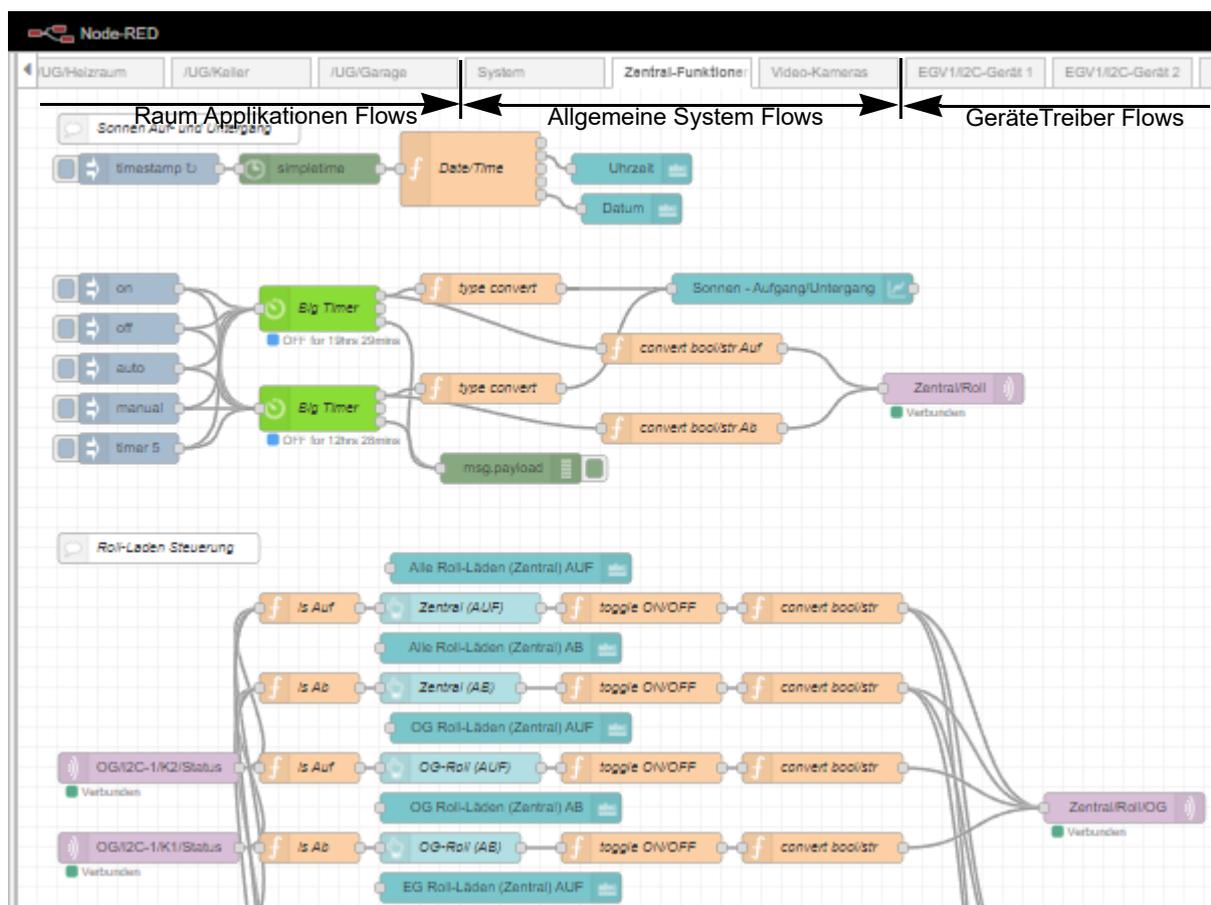


ABBILDUNG 15: Beispiel Flow für Zentralfunktionen Roll-Laden Steuerung Sonnen Auf-/Untergang

in Abbildung 15 auf Seite 22 dargestellt.

3.4.1 Programmier- und Bedienoberflächen

Organisatorisch ist im Node-RED Server das Programm Flow Browser Tab (Port 1880) und das Dashboard Browser Tab (Port 1880/ui) unabhängig voneinander. Über eine Passwort Zugang kann das Node-RED Programm Flow Tab für den Programmier mit exklusiven Zugangsrechten versehen werden. Das Dashboard GUI Tab passt sich in der Anzeigen Darstellung der grafischen Objekte dem Format des Endgerätes an.

Touch Screen Endgeräte wie Smartphones oder Tablets mit unterschiedlichen Anzeige Flächen Formaten sind in den Browser Einstellungen entsprechend einzustellen.

4.0 Dokumentation

Der Aufbau und die Logik der Dokumentation beruht darauf, dass jedes Gerät, jeder Sensor und Aktor im Namensraum des Gebäudes eindeutig durch den Ort und die Funktion eindeutig definiert ist. Mit der Logik der Namen Definition im Namensraum des Gebäudes werden die Datenobjekte für die Beschreibung der Zustände der Geräte, Aktoren und Sensoren in den Node-RED Flows exakt identisch mit den Topic-Namen bezeichnet. Mit dem MQTT-Protokoll wird die Information mit den ‚subscribe‘ - und ‚publish‘-Funktionen vom MQTT-Broker mit der Topic-Namen Adressierung transportiert. Mit den Node Verbindungen in den Node-RED Flows sind die logischen Verknüpfungen selbst erklärend. Durch geeignete Strukturierung der Flow Gestaltung lassen sich z.B. die Bindungen der Sensoren mit den Aktoren direkt aus dem Flow Programm-Code im Bindungen Flow ablesen. Es erübrigt sich eine zusätzliche Dokument Erstellung. Die Logik der Bildung des **Topic-Namen String** ist mit dem folgenden Schema nachvollziehbar. Die einzelnen Bestandteile des **Topic-Namen Strings** werden gemäß der Vorgaben des Namensraums des Gebäudes durch eine Verkettung mit dem ‚/‘ - Zeichen aneinandergereiht.

Das **Topic-Namen String**:

‚Wolfer13/EG/Erker/Licht-1‘

adressiert z.B. ein Datenobjekt im Gebäude **Wolfer13**, das im **EG**-Geschoss im Raum **Erker** den Licht Zustand des **Licht-1** repräsentiert.

In der Abbildung 16 auf Seite 24 ist das Flow sheet des **Geräte Treiber Flows** EG/I2C-1 als Beispiel Dokumentation dargestellt. Die Variablen der Datenpunkte, die mit den Topic-Namen Strings referenziert sind beschreiben direkt die Zustände der Geräte Anschlüsse.

In der Abbildung 17 auf Seite 25 sind im **Bindungen Flow** des Gerätes EG/I2C-1 die Bindungen der angeschlossenen physischen und virtuellen Sensoren des Raumes EG/ Erker gemäß des Leitungsplans realisiert. Die Bindungen der Status Ausgänge des Gerätes werden an den Eingängen der virtuellen Sensoren angeschlossen. Es müssen keine zusätzlichen Bindelisten geführt werden. Die Dokumentation entspricht also immer der aktuelle aktiven Software Variante. Am dargestellten Beispiel sind die Bindungen der Licht-Funktionen und der Roll-Laden Funktionen des Raumes E-Wohnung im UG-Geschoss zu entnehmen.

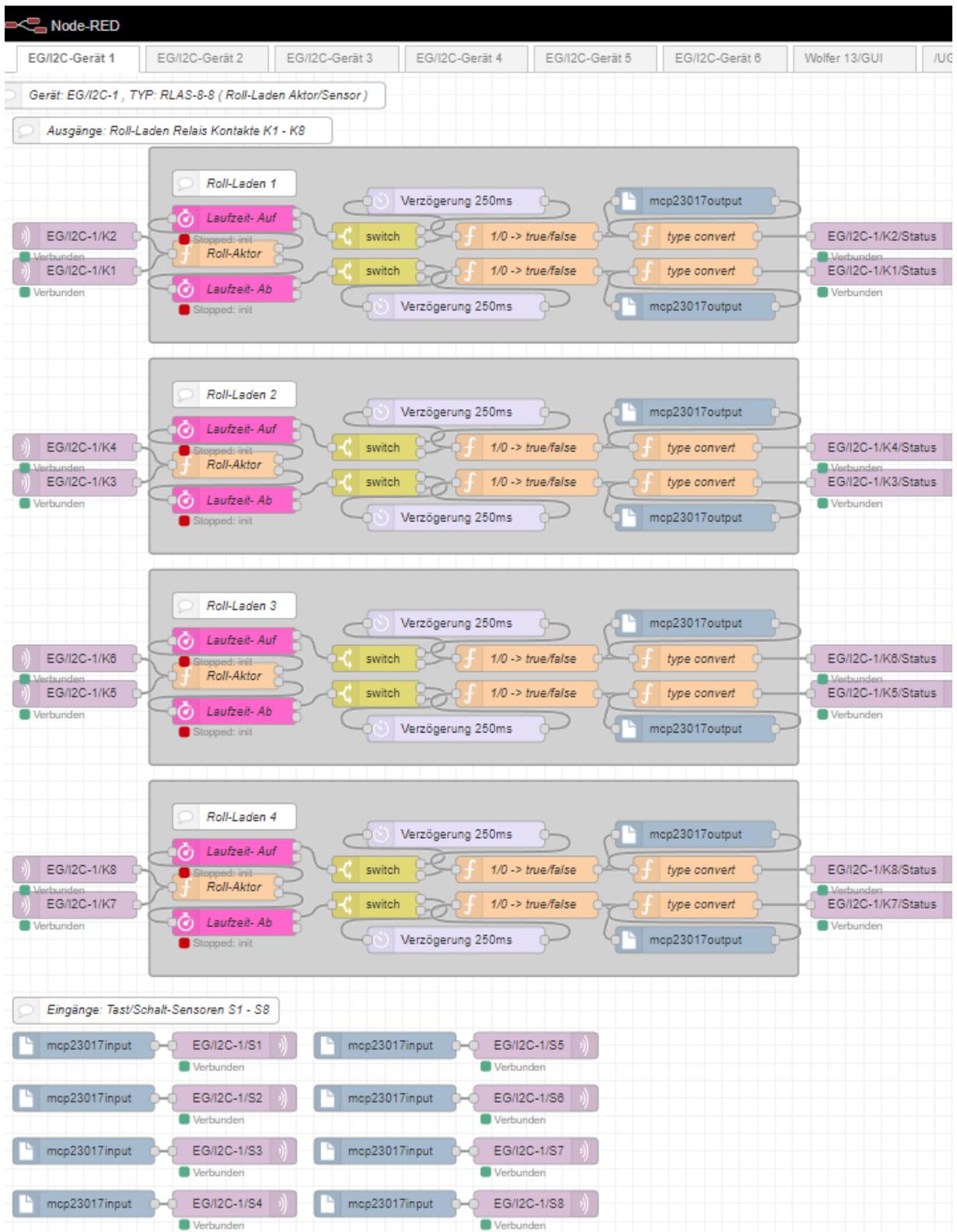


ABBILDUNG 16: Dokumentation Beispiel Flow-Sheet Geräte Treiber Flow EG/I2C-1

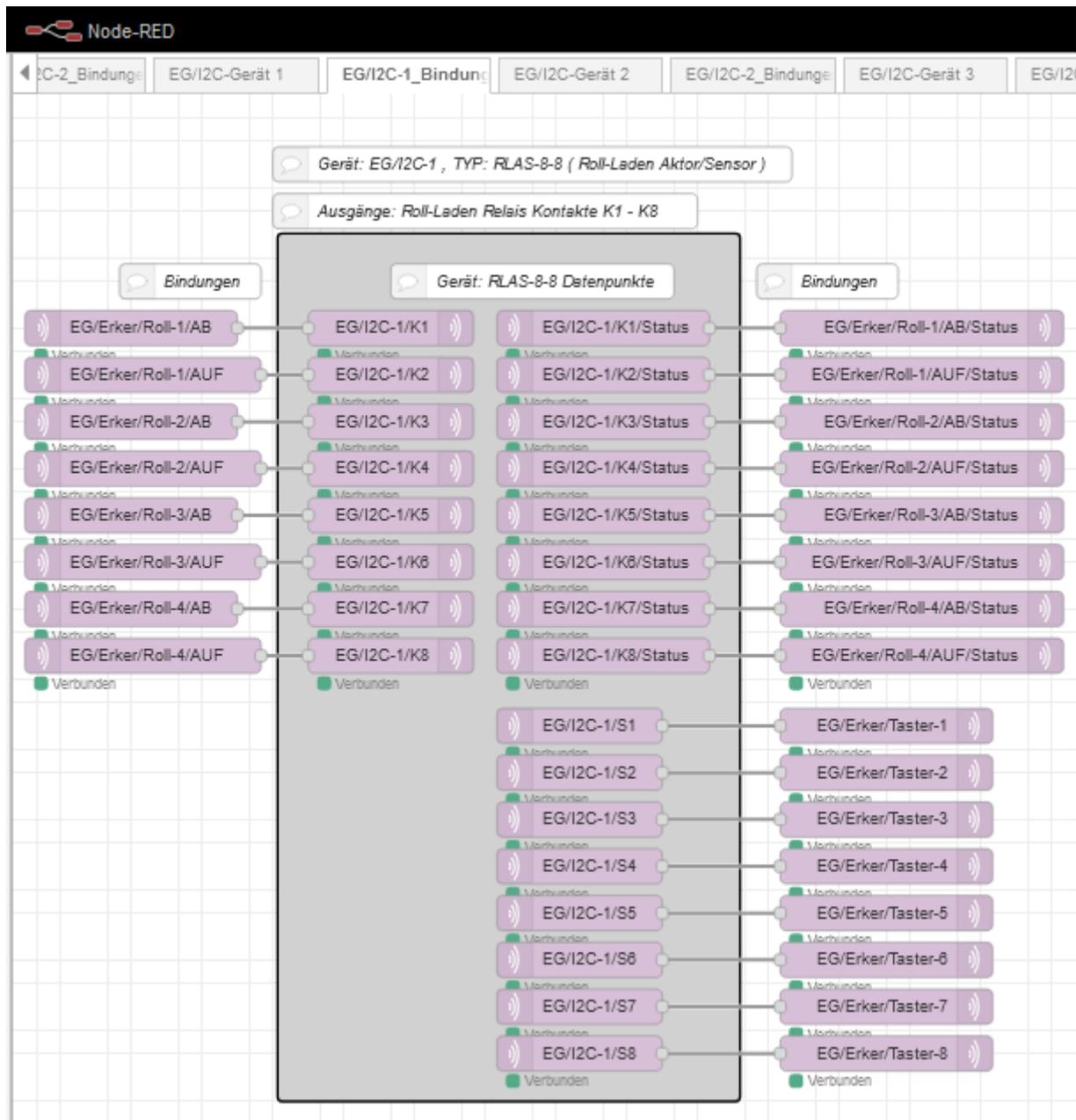


ABBILDUNG 17: Bindungen Flow zum Geräte Treiber Flow EG/I2C-1

Im **Raum Applikation Flow** am Beispiel des Raums UG/E-Wohnung in Abbildung 18 auf Seite 26 sind die Aktor Status Rückkopplungen an den Eingängen der virtuellen Sensoren angeschlossen. In den Raum Applikationen lassen sich die Bindungen der Raum Sensor Elemente mit den Raum Aktor Elementen direkt mit dem Raum Namen Definitionen vornehmen. Man muß bei der Codierung der Raum Applikationen also nicht wissen an welchen physischen Gerät im Verteiler die entsprechenden Anschlüsse realisiert sind. Mit der Konfiguration der Bindungen Flows ist die Zuordnung gemäß der Verdrahtung erfolgt. Die Codierung von Geräte Treiber Software und Raum Applikation Software ist damit sauber getrennt und dokumentiert.

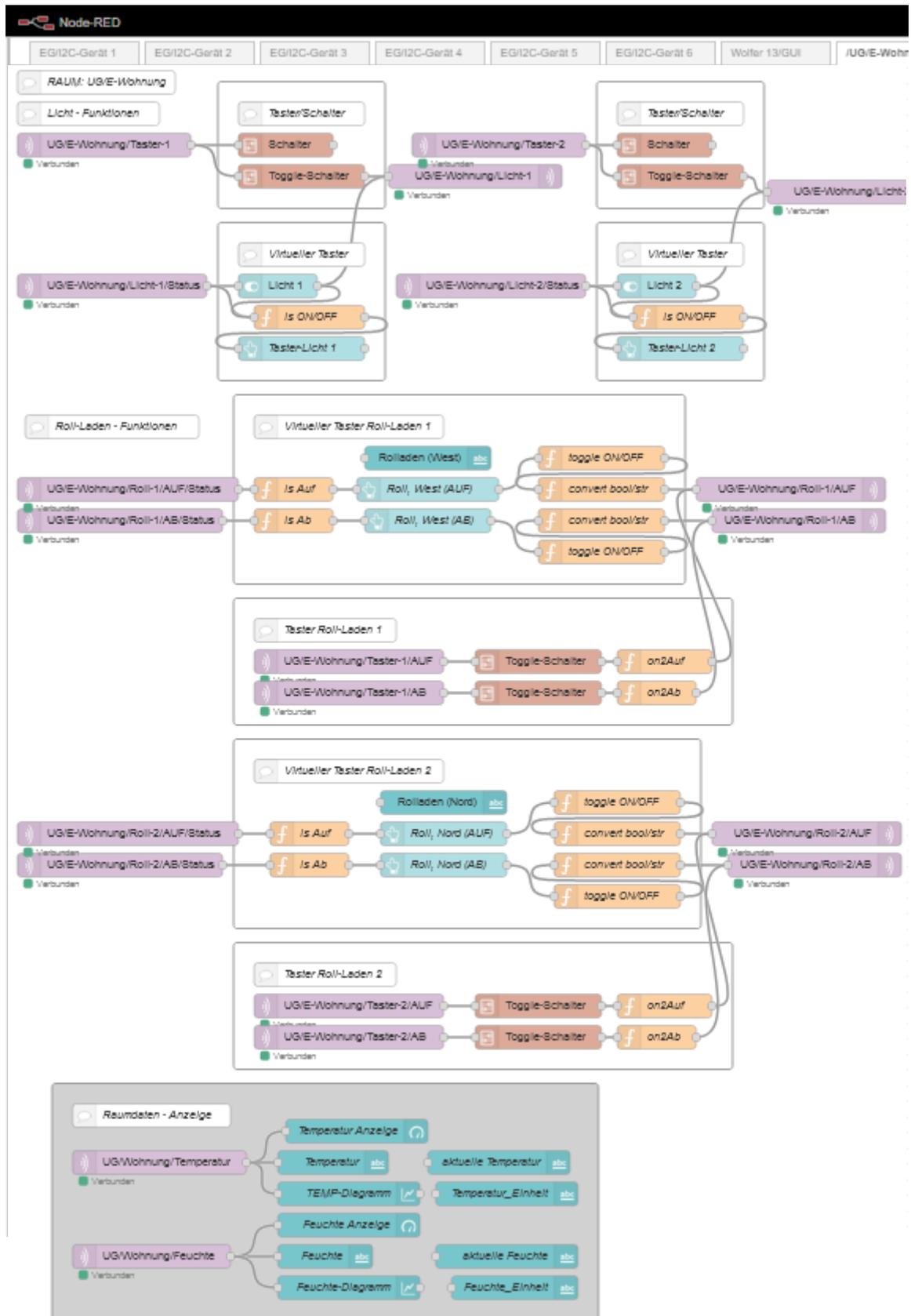
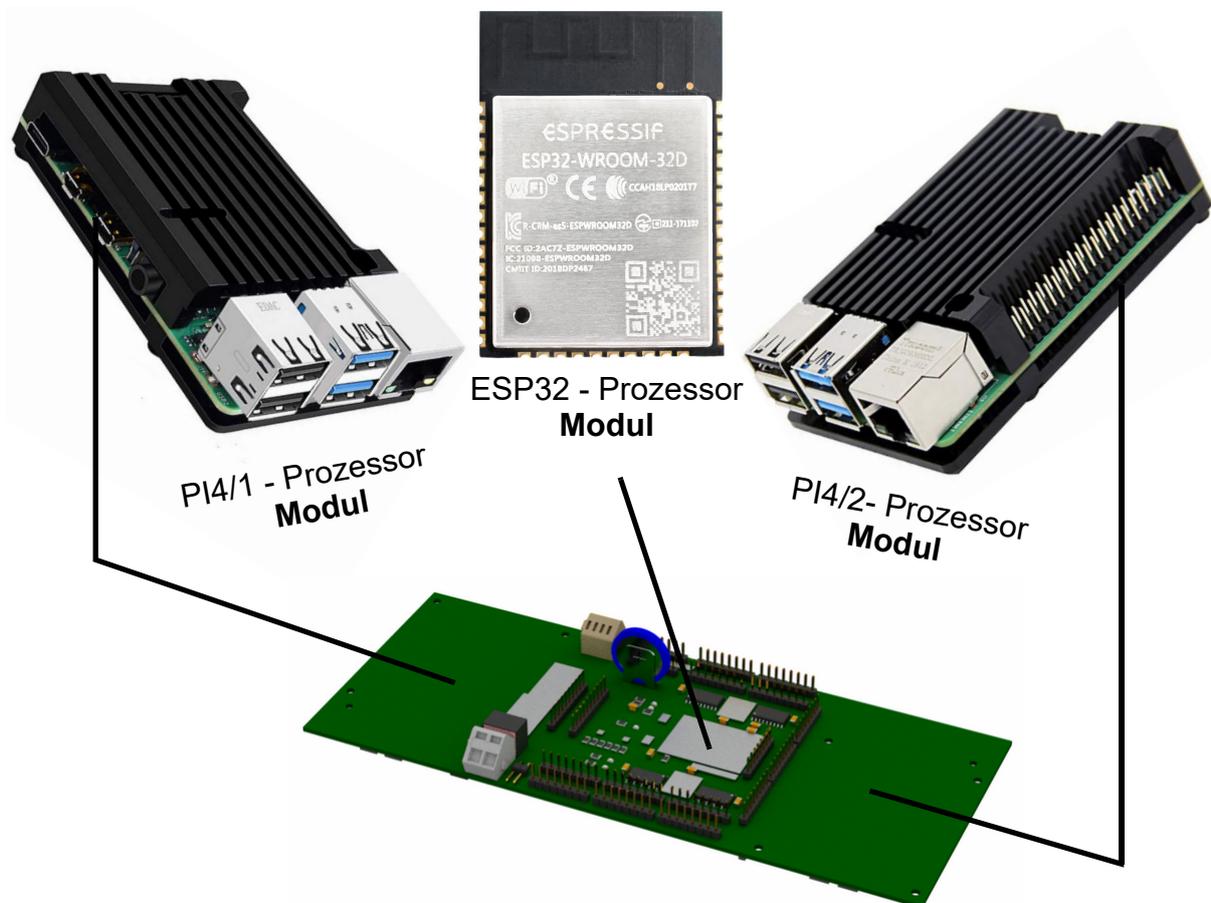


ABBILDUNG 18: Dokumentation Beispiel Raum-Applikation Flow UG/E-Wohnung Licht Funktionen, Roll-Laden Funktionen, Raumdaten - Anzeige

W SmartHome Open

2PI4/ESP32 - Server





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

2PI4/ESP32 - Server

1.0 Hardware Beschreibung

Der **2PI/ESP32 - Server** besteht aus einem Mehrprozessor-System mit 2 Raspberry PI4 Prozessor Modulen und einem ESP32 - WROOM Prozessor Modul. In Abbildung 1 auf Seite 30 ist der Aufbau im Blockschaltbild dargestellt. Damit ist ein Hardware- und Software redundantes SmartHome System realisierbar. Die Ausfall-Wahrscheinlichkeit kann damit extrem reduziert werden. Es wird eine Zuverlässigkeit erreicht, die in der SmartHome Automatisierung Maßstäbe setzt und ein Alleinstellungsmerkmal darstellt.

Hardware Eigenschaften:

- Galvanische Trennung aller Prozessoren und aller Leistungstreiber des System-GND.
- Hardware Watchdog-Überwachung mit Hardware Fail Detection
- Unabhängige Spannungsversorgungen,
auf die 3 Phasen des 230V-Versorgungsnetz verteilbar

Prozessor-Module

- 2 **Raspberry - PI4** Prozessor Module: ==> Hardware gleich; passives Kühlgehäuse

Spezifikationen pro Modul:

- Speicher: 2-, 4- oder 8-GByte RAM (Optional)
128 GByte SSD RAM-Disc (optional erweiterbar 257GB, 512GB, ...)
keine MicroSD-Speicherkarten im Einsatz
- Interfaces:
 - LAN- SmartHome Subnetz
 - RS 232 / RS485 Serielles Interface
 - I2C - Bus (galvanisch getrennt)
 - SPI - Bus (galvanisch getrennt)

- 1 **ESP32 - WROOOM32E16** Prozessor Module

- Speicher: 16 MB SPI flash RAM
- Frequenz: 240 MHz
- Interfaces:
 - WLAN- SmartHome Subnetz
 - RS 232 Serielles Interface (==> zur Programmierung)
 - I2C - Bus (galvanisch getrennt)
 - SPI - Bus (galvanisch getrennt)
 - 2 x 8 Output Ports zur Optokoppler Ansteuerung
(optional: Triac- und/oder PWM- Leistungstreibersteuerung)

Leistungstreiber - Interfaces (==> Digital - I/O)

- Versorgungsspannung: 24V mit Bezug auf System-GND für Leistungstreiber

- Output-Ports

- 64 Kanäle (8 x 8 Ports ==> Relaisansteuerung über N-Kanal MOSFET Leistungstreiber)

- Input-Ports (Spannungs-Eingänge)

- 64 Kanäle (8 x 8 Ports ==> Digitale Sensor Eingangs-Signale:
Pegel: 0 - 25V
Treshold: 2,8V (false <==> true))

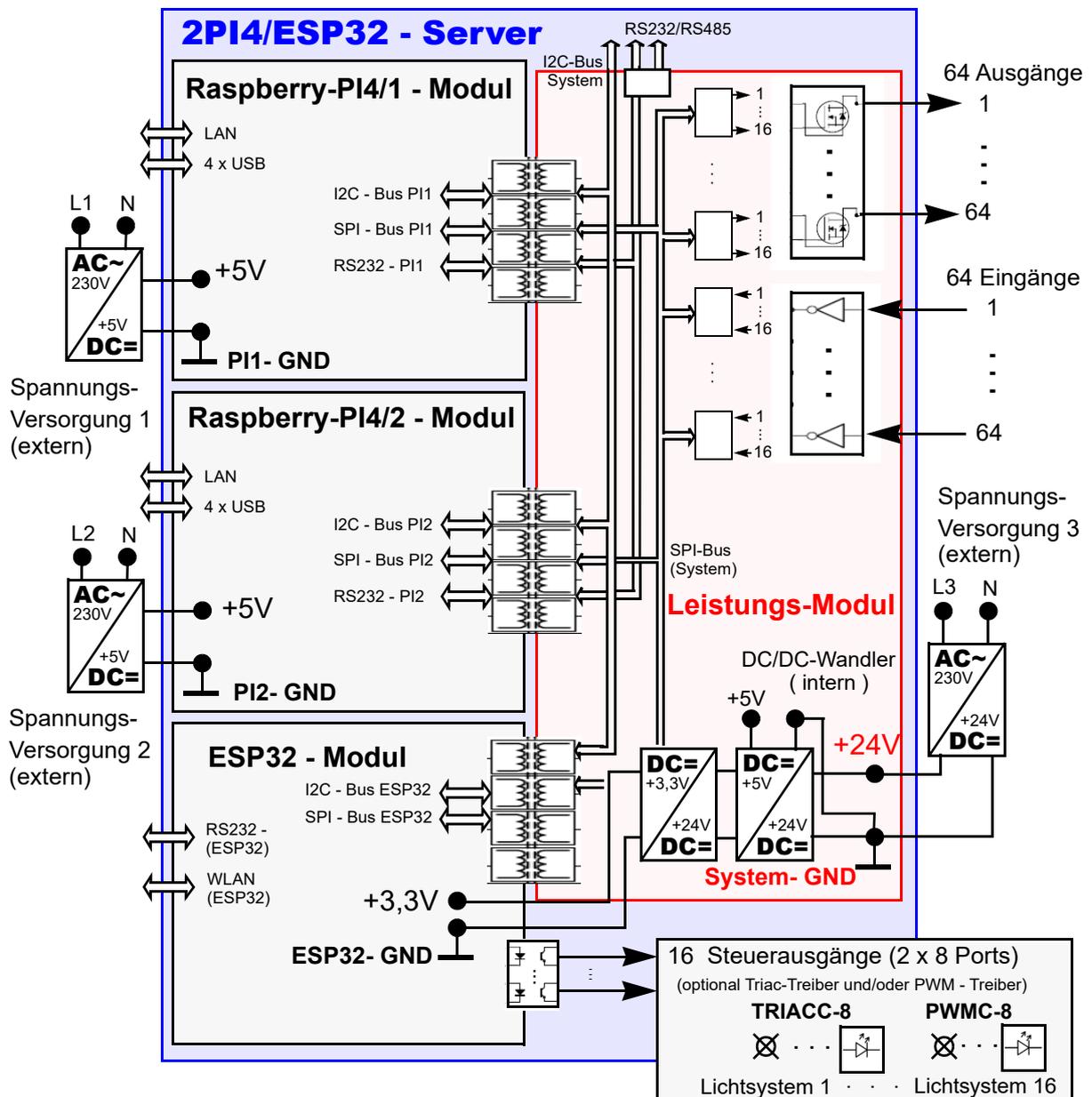


ABBILDUNG 1: Blockschaltbild Aufbau 2PI4/ESP32-Server



ABBILDUNG 2: 2PI/ESP32 - Server

W SmartHome Open

2PI_Zero/ESP32 - Server





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

2PI_Zero/ESP32 - Server

1.0 Hardware Beschreibung

Der **2PI_Zero/ESP32 - Server** ist eine Low Power Hardware Variante und besteht aus einem Mehrprozessor-System mit 2 Raspberry PI-Zero 2 W Prozessor Modulen und einem ESP32 - WROOM Prozessor Modul. Alternativ können als Prozessor-Module auch ein Raspberry PI2, 3 oder 4 und ein PI-Zero 2 W gewählt werden. In Abbildung 1 auf Seite 37 ist der Aufbau im Blockschaltbild dargestellt. Damit ist ein Hardware- und Software redundantes System realisierbar, das optimal als Geräte-Treiber Server in einem Etagen Verteiler eingesetzt werden kann.

Hardware Eigenschaften:

- Galvanische Trennung aller Prozessoren und aller Leistungstreiber des System-GND.
- Hardware Watchdog-Überwachung mit Hardware Fail Detection
- Unabhängige Spannungsversorgungen,
auf die 3 Phasen des 230V-Versorgungsnetz verteilbar

Prozessor-Module

- 2 **Raspberry - Zero 2 W** Prozessor Module: ==> Hardware gleich; jeweils passives Kühlgehäuse

Spezifikationen 1. Modul:

- Speicher: 0,5-GByte RAM
128-GByte SSD RAM-Disc (optional erweiterbar 257GB, 512GB, ...)
keine MicroSD-Speicherkarten im Einsatz

Spezifikationen 2. Modul:

- Speicher: 0,5-GByte RAM
32-GByte MicroSD-Speicherkarten im Einsatz
- Interfaces: - WLAN- SmartHome Subnetz
- I2C - Bus (galvanisch getrennt)

- 1 **ESP32 - WROOM32E16** Prozessor Module

- Speicher: 16 MB SPI flash RAM
- Frequenz: 240 MHz
- Interfaces: - WLAN- SmartHome Subnetz
- RS 232 Serielles Interface (==> zur Programmierung)

- I2C - Bus (galvanisch getrennt)
- 2 x 8 Output Ports zur Optokoppler Ansteuerung
(optional: Triac- und/oder PWM- Leistungstreibersteuerung)

- Versorgungsspannungen

1. Modul:

Externes 5V-Netzteil über USB C - Stecker galvanisch getrennt von System-GND

2. Modul:

internes DC/DC- Netzteil 24V/5V galvanisch getrennt

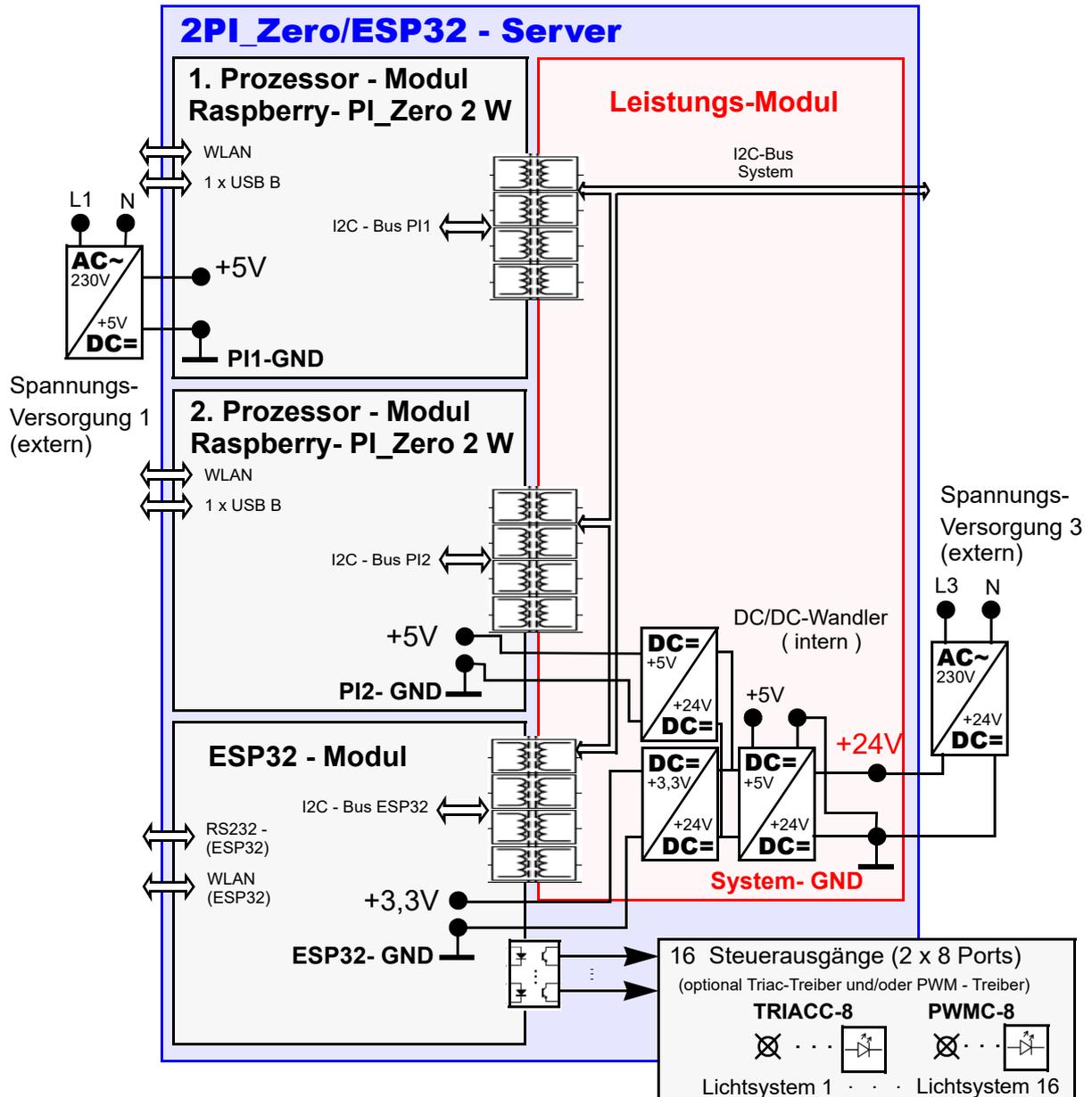


ABBILDUNG 1: Blockschaltbild Aufbau 2PI_Zero/ESP32-Server

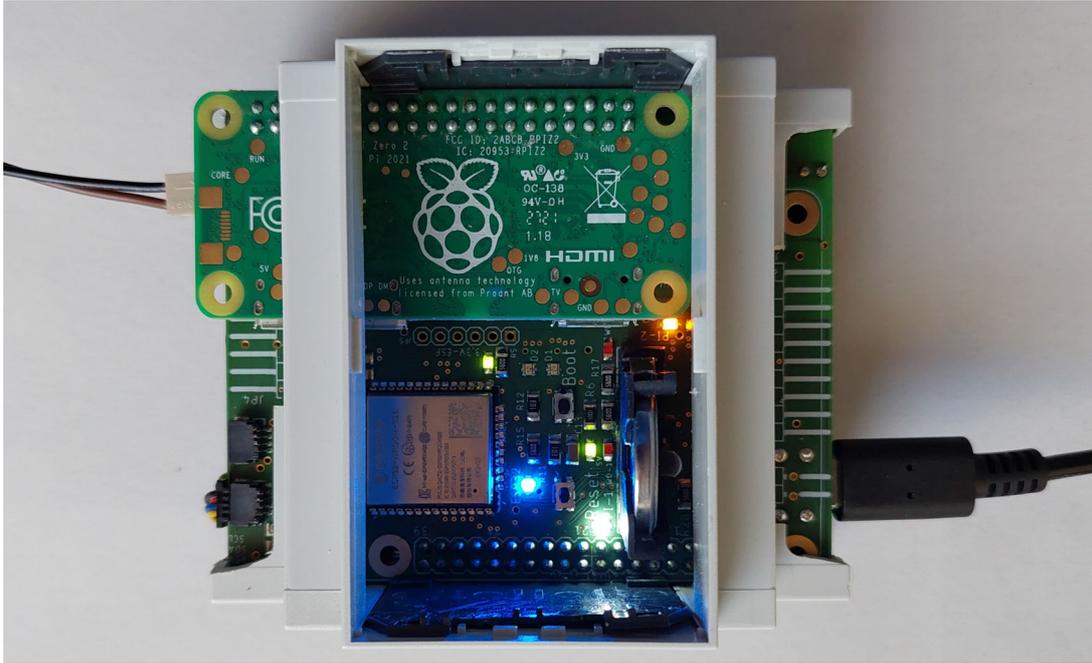


ABBILDUNG 2: 2PI_Zero/ESP32 - Server

W *SmartHome* *Open*

Schalter Aktor/Sensor





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

Schalter Aktor/Sensor

1.0 Hardware Beschreibung

Funktion: **8 Schaltausgänge** an Arbeitskontakten,
8 Schalteingänge

Bezeichnung: **SAS-8-8**

Schnittstellen

Kommunikation: **I2C - Bus**, adressierbar über 3-poligen DIP-Schalter

2 x Stecker **Sparkfun qwiic**
(=> Daisy Chain, Hot Swap - fähig)

Ports

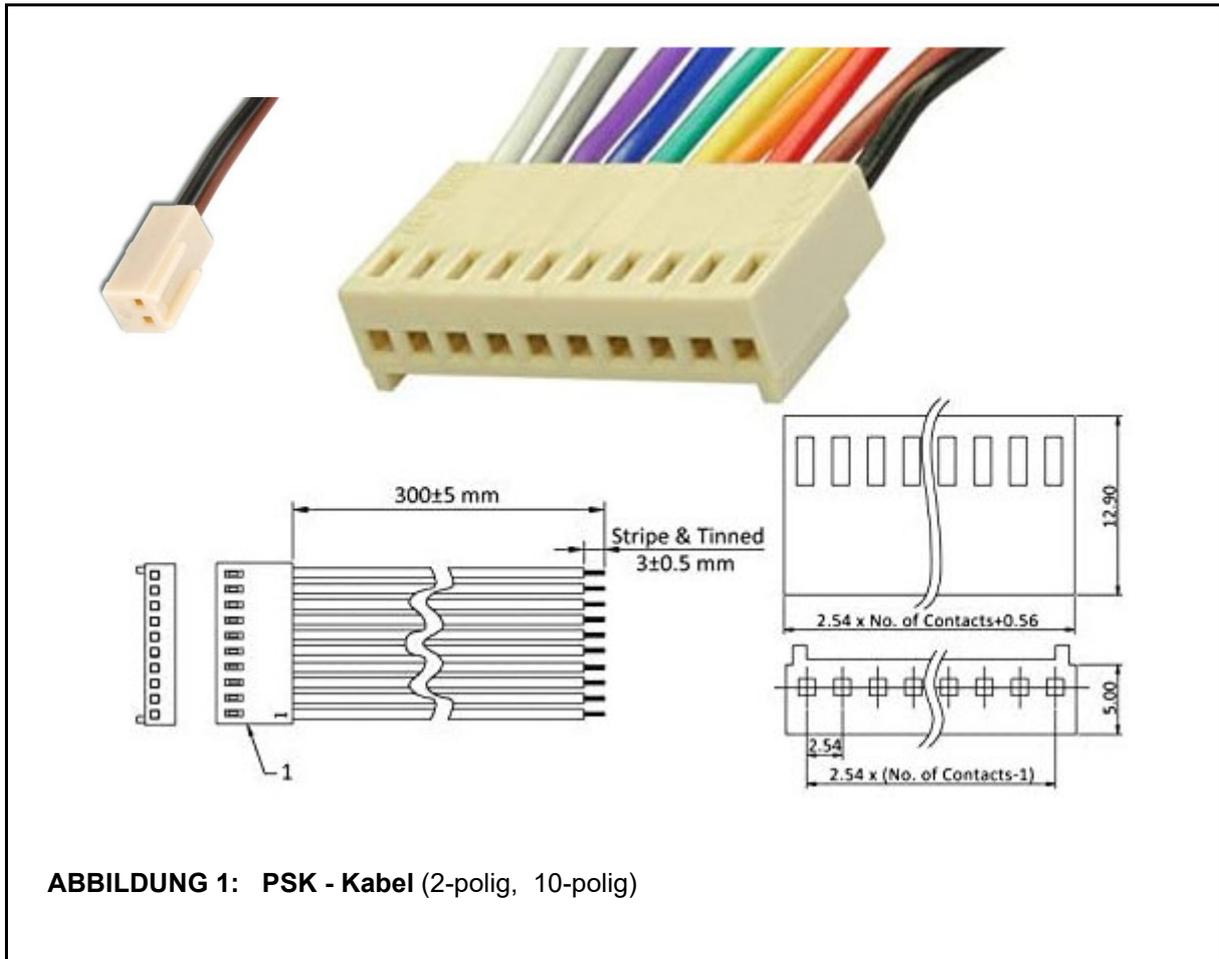
Inputs: 8
 Spannung 0 - +30V_{DC} , Schaltschwelle $V_{ON} = 2,8V$
 Verbindung: PSK-Stecker (10-polig)
 Spannungsversorgung: PSK-Stecker (2-polig) => (+24V, GND)
 (Abbildung 1 auf Seite 42)

Outputs: 8
 NPN-Darlington Transistor Leistungstreiber Endstufen mit clamp Dioden
 Strom: 500 mA (max)
 Versorgungsspannung: $V_{CE} = +24 V$ (+30V max)
 Verbindung: PSK-Stecker (10-polig)
 (Abbildung 1 auf Seite 42)

Ausgänge für induktive Lasten ausgelegt. Ideal geeignet zur Ansteuerung z.B. der Relais-Module **RKS-8** und **RKW-8** .

Eigenschaften

- Ausgänge jeweils manuell steuerbar über 8 Schalter
- Zustandsanzeige der Ausgänge jeweils über 8 LED (rot)



2.0 Software Beschreibung

Gerätetreiber über **Node-RED** ==> (Abbildung 2 auf Seite 43)

Input Device-node:

- 8 boolsche Datenpunkt Variablen: - Zustand true/false,

Output Device-node:

- 8 boolsche Datenpunkt Variablen: - Zustand true/false.

Kommunikationsprotokoll

MQTT: Subscribe / Publich - Functions mit Topic- Adressierung. Topic-Namensraum abbildbar auf Geräteinstallation sowie Gebäude- und Raumstruktur.

Kommunikationsmanagement über **Mosquitto-Broker** (Server-Funktion)

Applikationsfunktionen über strukturierte **Node-RED Flows** realisierbar

Grafische Visualisierung erfolgt über **Dashboard-Anzeige** Elemente

Programm Beispiele: Abbildung 2 auf Seite 43, Abbildung 3 auf Seite 44

ABBILDUNG 2: Node-RED Device Flow Schalter Aktor

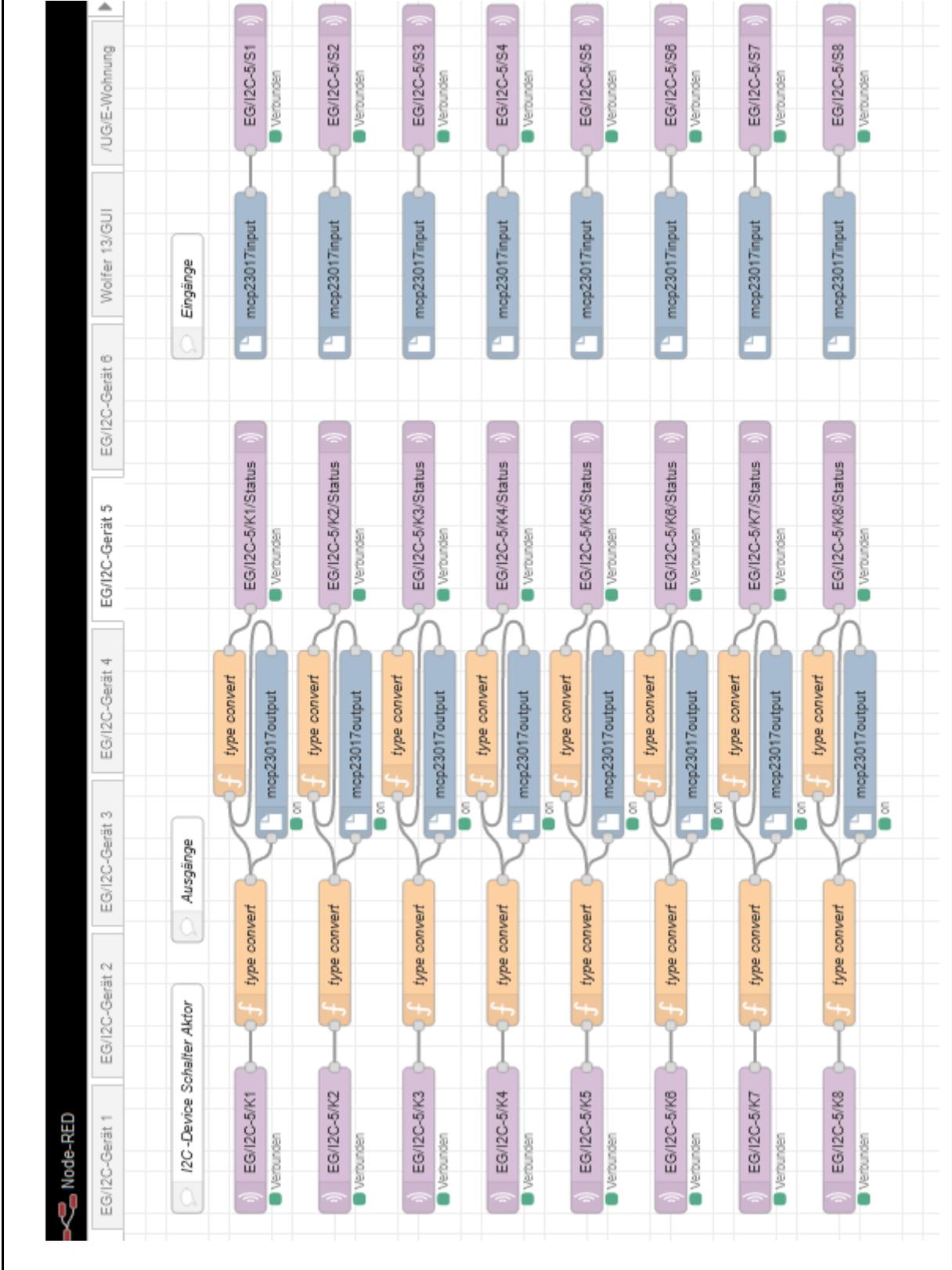
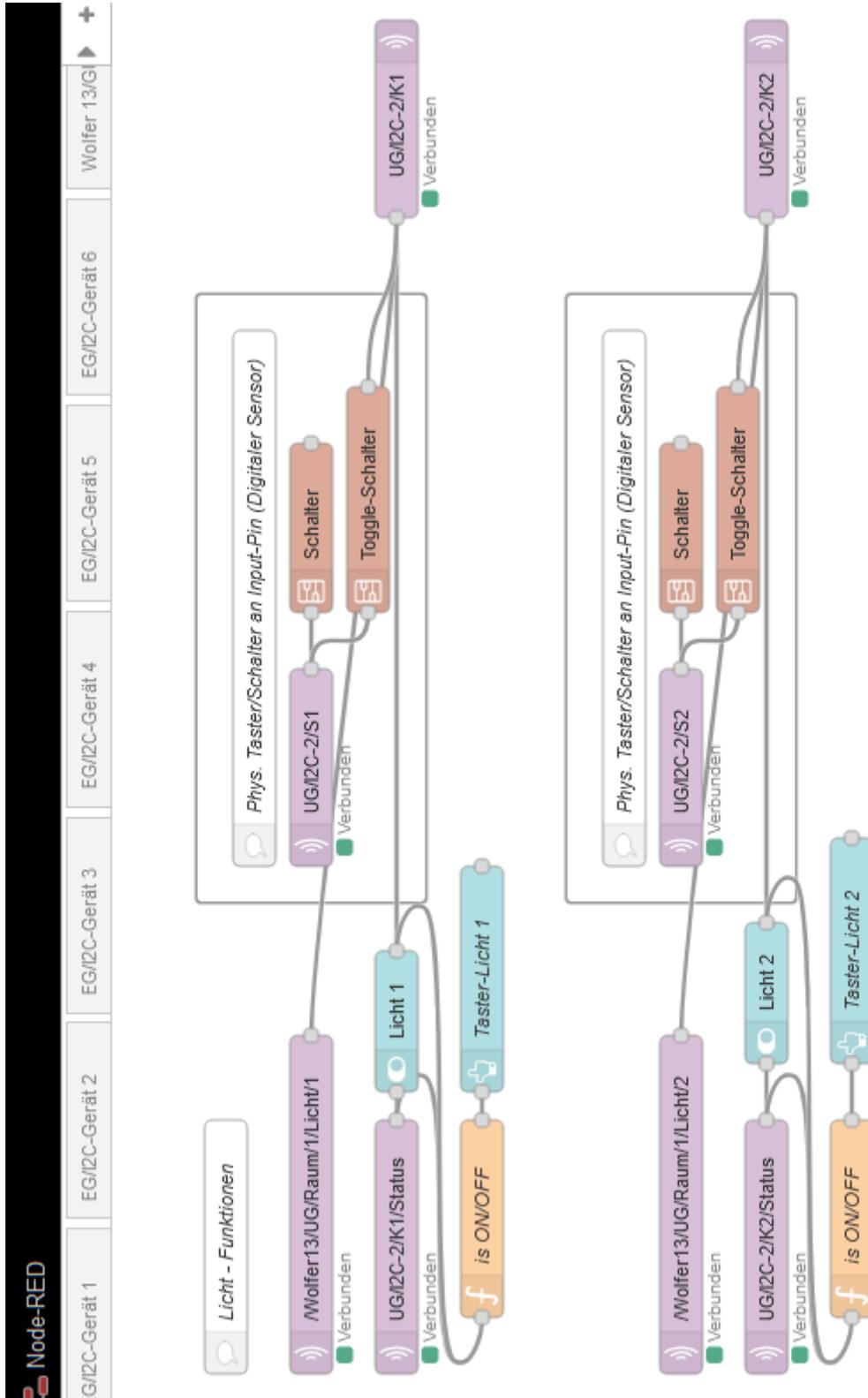


ABBILDUNG 3: Node-RED Application Flow Licht - Schalter

Node-RED



3.0 Anschluss Belegung

In Abbildung 4 auf Seite 45 ist das **Schalter Aktor/Sensor Modul SAS-8-8** in Kombination mit dem Relais Modul **Schließer RKS-8** im Hutschienen Gehäuse mit den Klemmen Bezeichnungen für die Installation dargestellt.



L1, L2, L3, L4, L5, L6, L7, L8: Versorgungsspannung 230V/AC pro Schalter
 K1, K2, K3, K4, K5, K6, K7, K8: Arbeitskontakt pro Verbraucher
 (Phasen-Leiter beliebig wählbar)
 N - Leiter ist am jeweiligen Verbraucher anzuschließen

ABBILDUNG 4: Schalter Aktor/Sensor Modul **SAS-8-8** mit Relais Modul **Schließer RKS-8** im Hutschienengehäuse

W *SmartHome* *Open*

Roll-Laden Aktor/Sensor





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

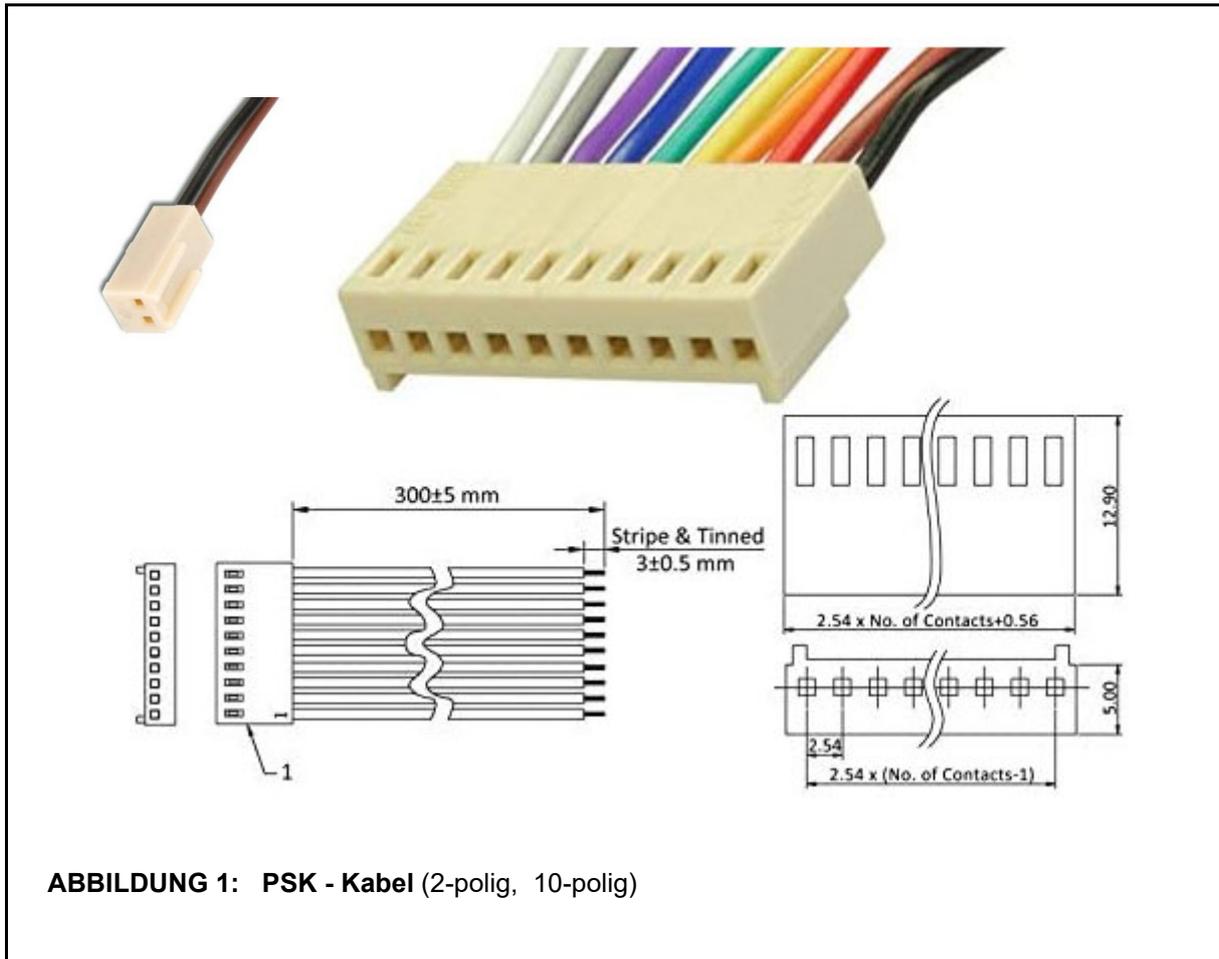


ABBILDUNG 1: PSK - Kabel (2-polig, 10-polig)

2.0 Software Beschreibung

Gerätetreiber über **Node-RED** ==> (Abbildung 2 auf Seite 51)

Input Device-node:

- 8 boolsche Datenpunkt Variablen: - Zustand true/false,

Output Device-node:

- 8 boolsche Datenpunkt Variablen: - Zustand true/false.

Kommunikationsprotokoll

MQTT: Subscribe / Publich - Functions mit **Topic**-Adressierung. Topic-Namensraum abbildbar auf Geräteinstallation sowie Gebäude- und Raumstruktur.

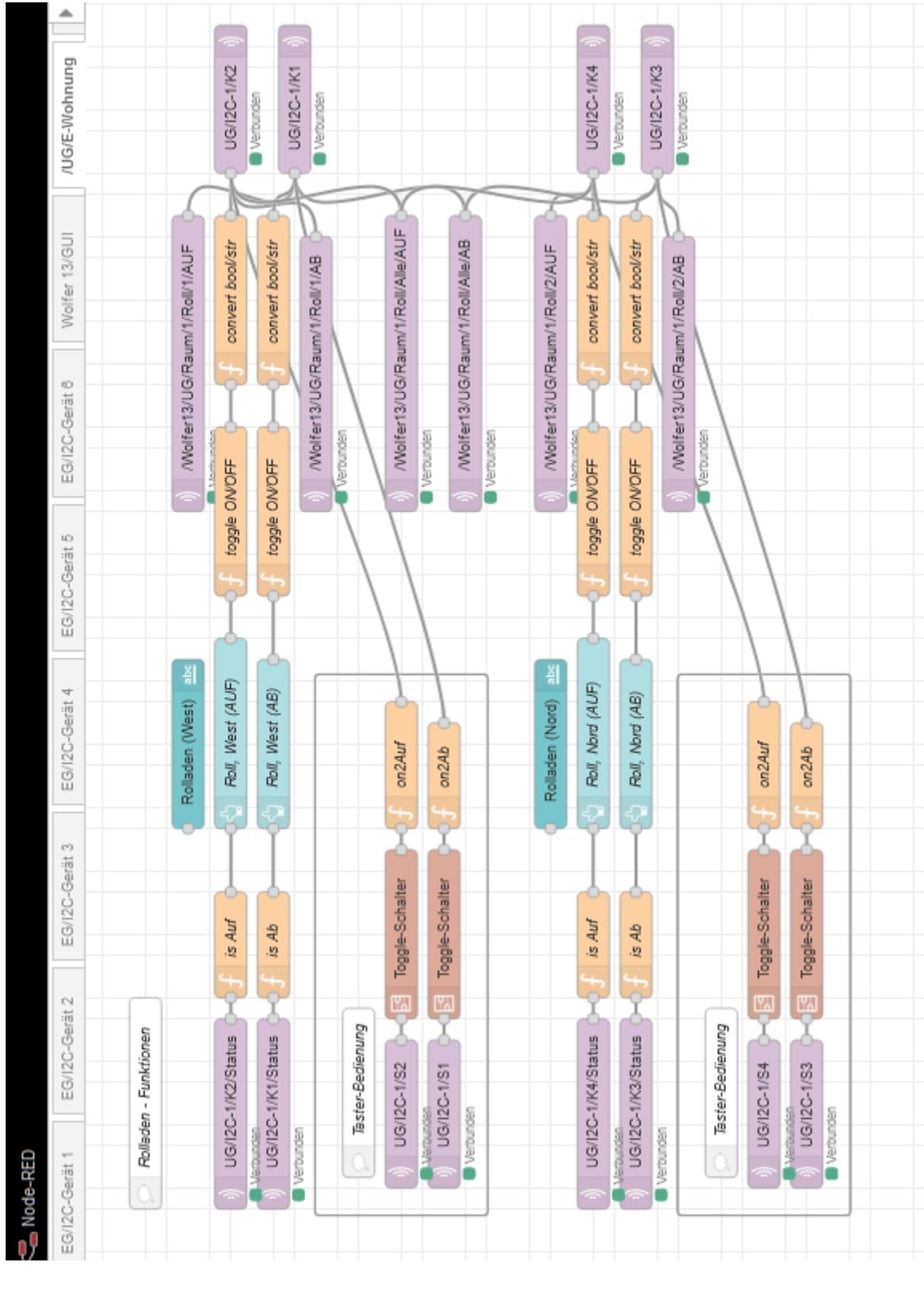
Kommunikationsmanagement über **Mosquitto-Broker** (Server-Funktion)

Applikationsfunktionen über strukturierte **Node-RED Flows** realisierbar

Grafische Visualisierung erfolgt über **Dashboard-Anzeige** Elemente

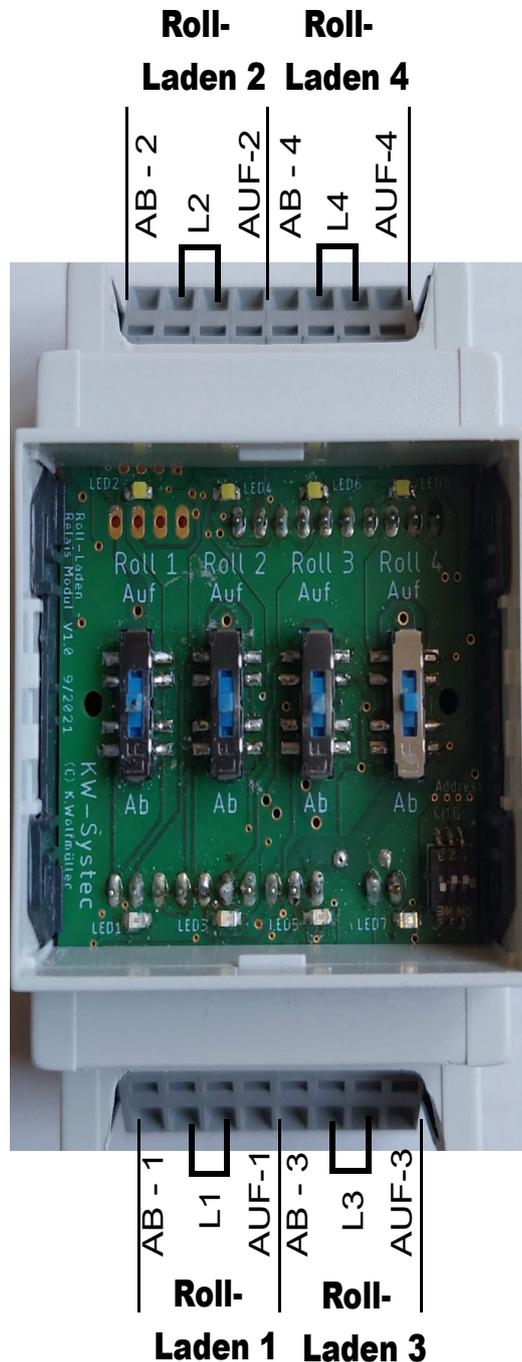


ABBILDUNG 3: Node-RED Application Flow Roll-Laden/Raum Beispiel



3.0 Anschluss Belegung

In Abbildung 4 auf Seite 53 ist das **Roll-Laden Aktor/Sensor Modul RLAS-8-8** in Kombination mit dem Relais Modul **Roll-Laden RKR-4** im Hutschienen Gehäuse mit den Klemmen Bezeichnungen für die Installation dargestellt



L1, L2, L3, L4: Versorgungsspannung 230V/AC pro Roll-Laden
(Phase pro Roll-Laden beliebig , Klemmen intern verbunden)

N - Leiter ist am jeweiligen Roll-Laden anzuschließen

ABBILDUNG 4: Roll-Laden Aktor/Sensor Modul **RLAS-8** mit Relais Modul **Roll-Laden RKR-4** im Hutschienengehäuse

W *SmartHome* *Open*

Dimmer/Motion - Controller



**Mosfet - Modul
PWMC-8**



**Triac - Modul
TRIACC-8**



KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

Dimmer/Motion - Controller

1.0 Hardware Beschreibung

Bezeichnung: **DMC-32**

Schnittstellen

Kommunikation: **I2C - Bus**, adressierbar über 3-poligen DIP-Schalter
WLAN

2 x Stecker **Sparkfun qwiic**
(=> Daisy Chain, Hot Swap - fähig)

Input:

Spannungsversorgung: PSK-Stecker (2-polig) => (+24V, GND)

Output - Ports

PWM: 16 Kanäle (Hardware Pulsbreiten-Generator)

Steuersignale: $u[i] = 0 - +5V_{DC}$, $i = 1, \dots, 16$, 25mA Stromsenke
Frequenz $f = 24 - 1526$ Hz, 12-bit Auflösung
(=> Treiberendstufen zur Optokoppler Steuerung)

Verbindung: 2x PSK-Stecker (10-polig) => Optokoppler-Signale
(Abbildung 1 auf Seite 58)

Anwendung: - Dimmer für $12V_{DC}$ LED-Lampen
- Motion-Control DC-Motoren

Phasenanschnitt: 16 Kanäle (Zündimpuls Generator zur Triac-Steuerung)

Steuersignale: 0 - $+5V_{DC}$ (=> Treiber zur Optokoppler Ansteuerung)
(=> Trigger über Zero-Detect, $230 V_{AC}$ Netzspannung)

Verbindung: 2x PSK-Stecker (10-polig) => Optokoppler-Signale
1x PSK-Stecker (2-polig), => zero-detect
(Abbildung 1 auf Seite 58)

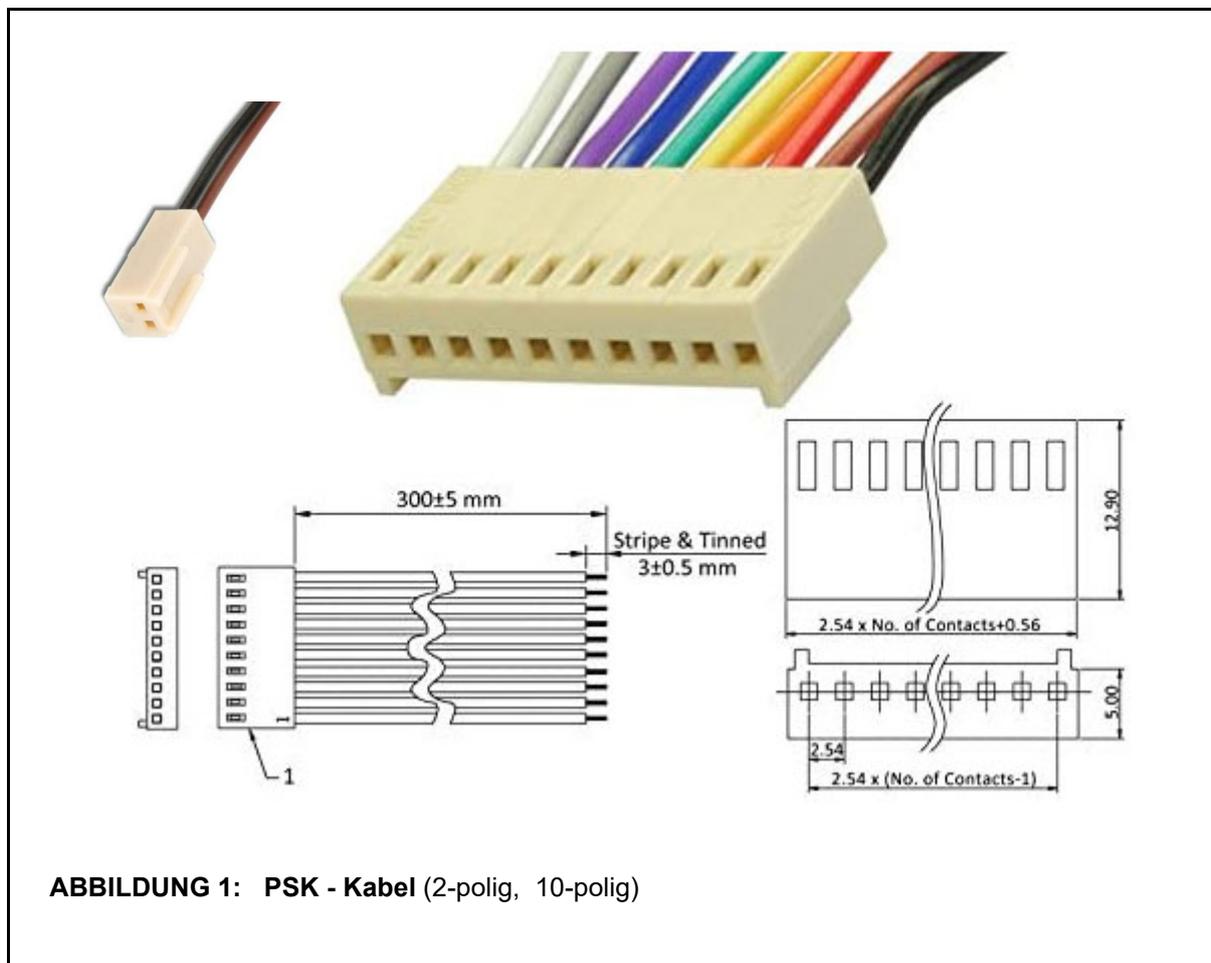
Anwendung: - Dimmer für $230V_{AC}$ LED-Lampen (dimmbar)
- Leistungssteuerung Wechselstrom-Verbraucher
(optionaler Betrieb: Signale programmierbar als PWM-Steuersignale)

Ausgänge kompatibel zu den Eingängen der Leistungsmodule

- PWM-Leistungstreiber Modul **PWMC-8** (Abbildung 2 auf Seite 59)
- Triac-Leistungstreiber Modul **TRIACC-8** (Abbildung 4 auf Seite 61)

Eigenschaften

- PWM-Zustandsanzeige je Kanal über 16 LED (rot)
- 2 Prozessor Statusanzeigen: Power ON > LED (rot) , Active > LED (blau)
- Kombinierbar mit PWM Leistungstreiber Modul **PWMC-8** oder Triac-Leistungstreiber Modul **TRIACC-8** im Hutschienengehäuse



1.1 PWM-Leistungstreiber Modul **PWMC-8**

In Abbildung 2 auf Seite 59 ist das Modul im Hutschienen Gehäuse dargestellt.

Spezifikation

Leistungsausgänge:	8	N-Kanal Mosfet
Spannung		$V_{DS} = 30V$ (max)
Strom		$I_D = 63 A$ (max)

Alle Spannungsversorgungen sind galvanisch getrennt.

PSK-Stecker (10-polig)
- Optokoppler Signale

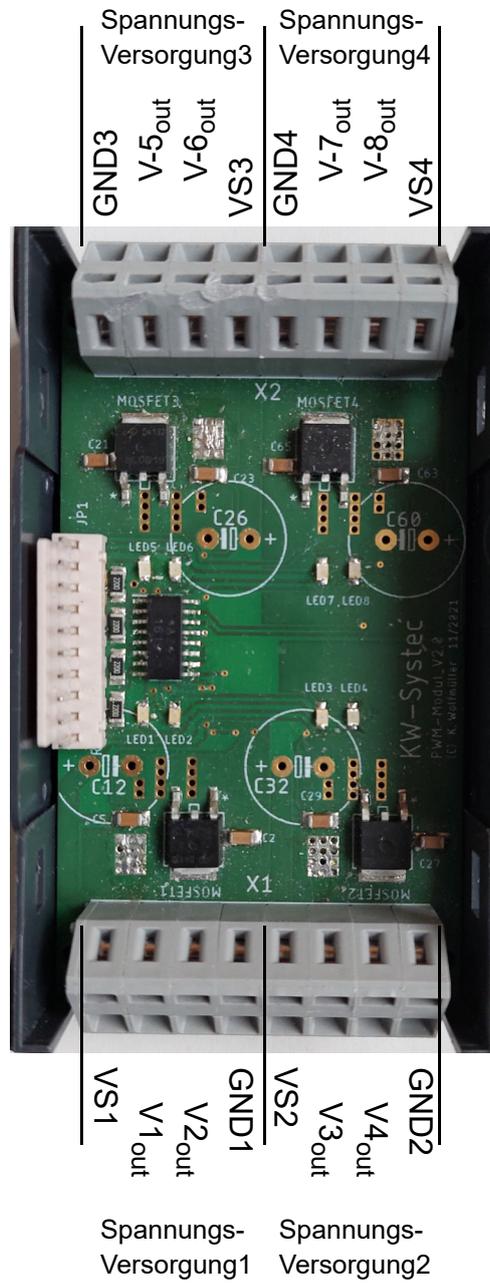


ABBILDUNG 2: PWM-Leistungstreiber Modul **PWWC-8**

PSK-Stecker (10-polig)
- Optokoppler Signale

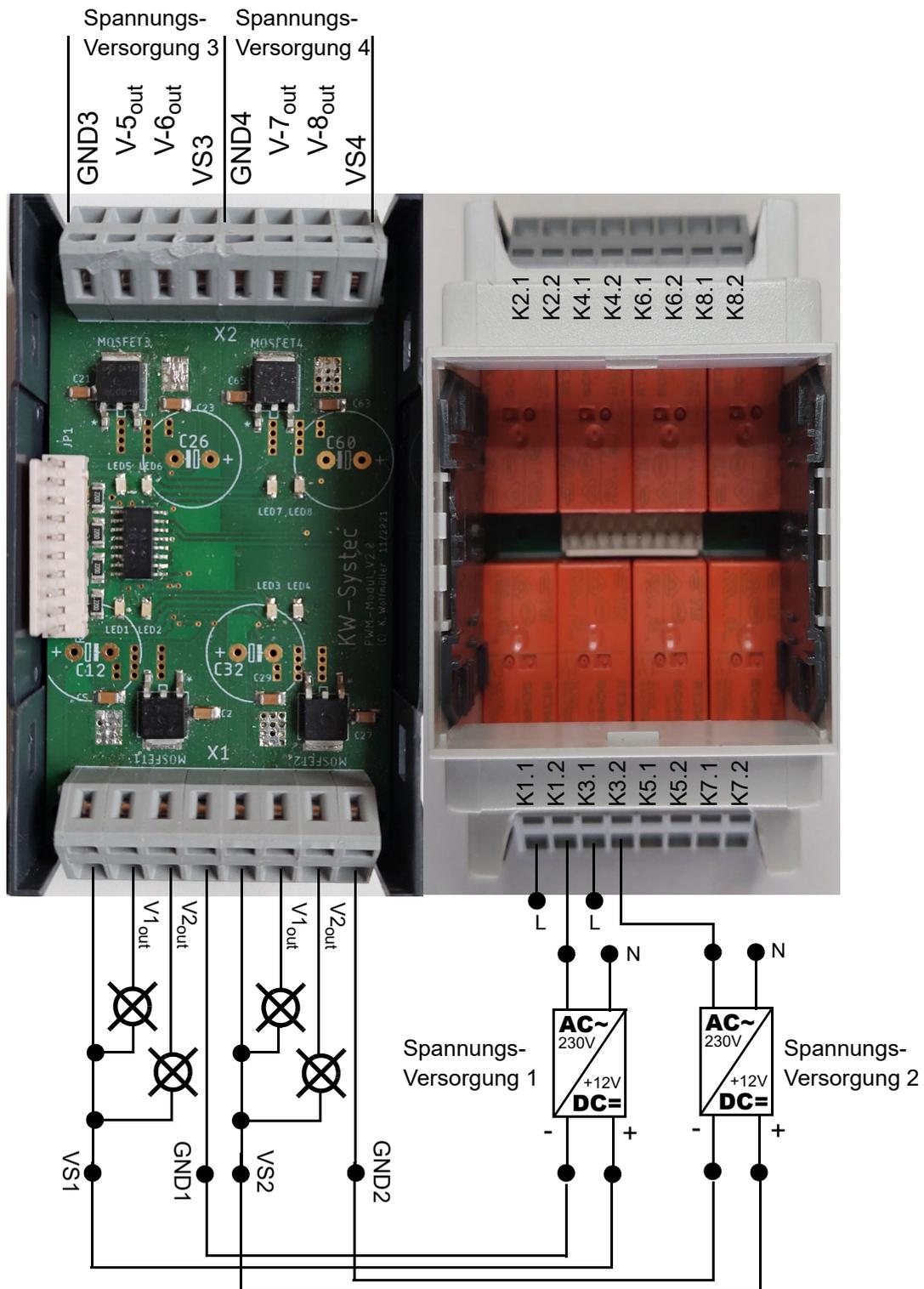


ABBILDUNG 3: Schaltungsvorschlag Lampen/Dimmer in Kombination mit Schalter durch PWM Leistungstreiber Modul **PWMC-8**, Relais Modul Schließer **RKS-8** und 12V_{DC}-Netzteile

1.2 Triac-Leistungstreiber Modul **TRIACC-8**

In Abbildung 4 auf Seite 61 ist das Modul im Hutschienen Gehäuse mit der Klemmen Belegung dargestellt.

Spezifikation

Leistungsausgänge:	8	Triac - Endstufen, Phasenanschnitt
Spannung		$V_{AC} = 230V\sim$
Strom		$I = 8 A (max)$

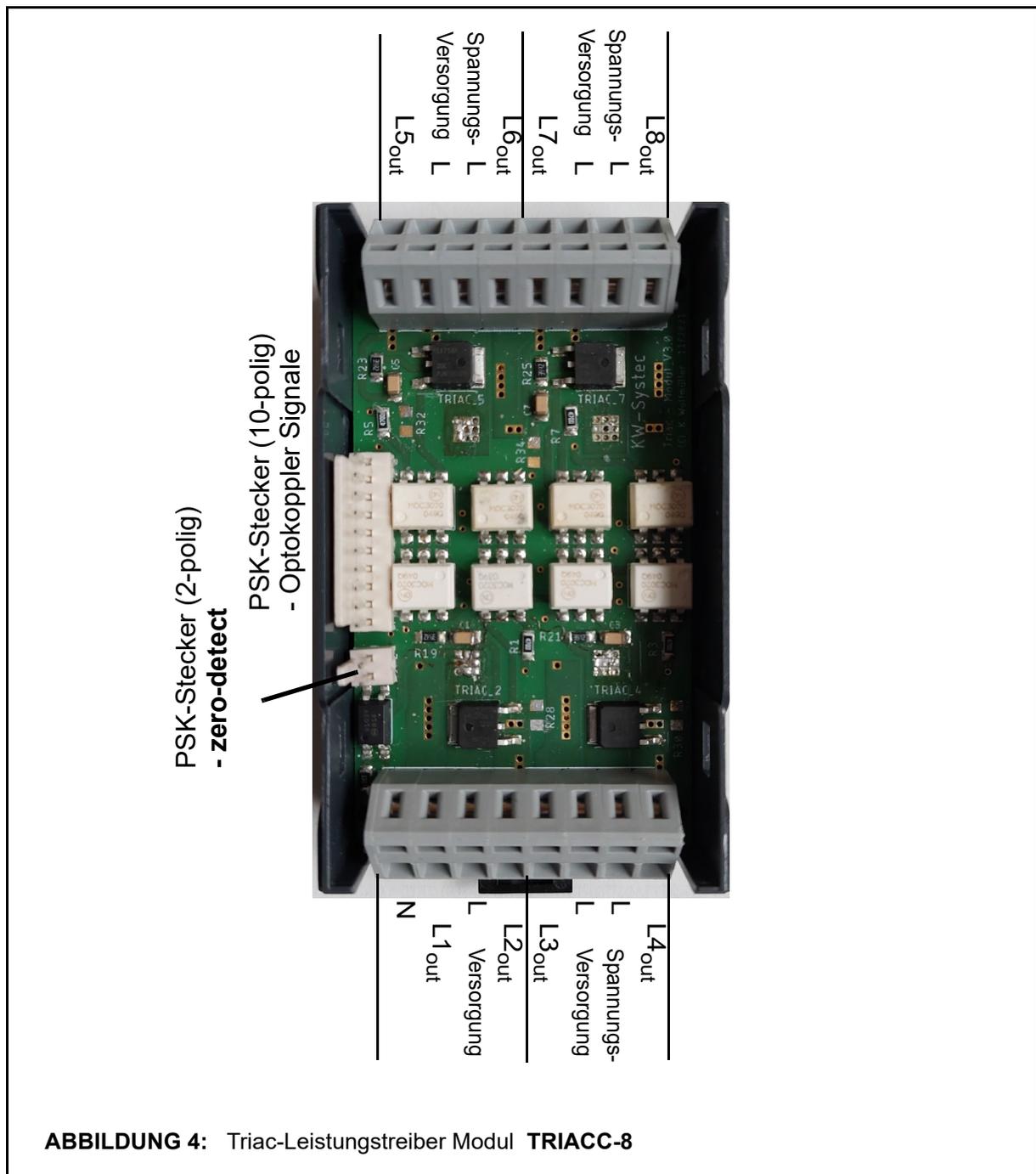


ABBILDUNG 4: Triac-Leistungstreiber Modul **TRIACC-8**

2.0 Software Beschreibung

Gerätetreiber über **Node-RED** ==> (Abbildung 5 auf Seite 63)

(Abbildung 6 auf Seite 64)

Input Device-node (PWM):

- 16 (int_16)-Datenpunkt Variablen: - Zustand aktueller Dimmwert (0 - 100%)

Input Device-node (TRIAC bzw optional PWM):

- 16 (int_16)-Datenpunkt Variablen: - Zustand aktueller Dimmwert (0 - 100%)

Kommunikationsprotokoll

MQTT: Subscribe / Publich - Functions mit **Topic**- Adressierung. Topic-Namensraum abbildbar auf Geräteinstallation sowie Gebäude- und Raumstruktur.

Kommunikationsmanagement über **Mosquitto-Broker** (Server-Funktion)

Applikationsfunktionen über strukturierte **Node-RED Flows** realisierbar

Grafische Visualisierung erfolgt über **Dashboard-Anzeige** Elemente



ABBILDUNG 5: Node-RED Device Flow PWM Aktor

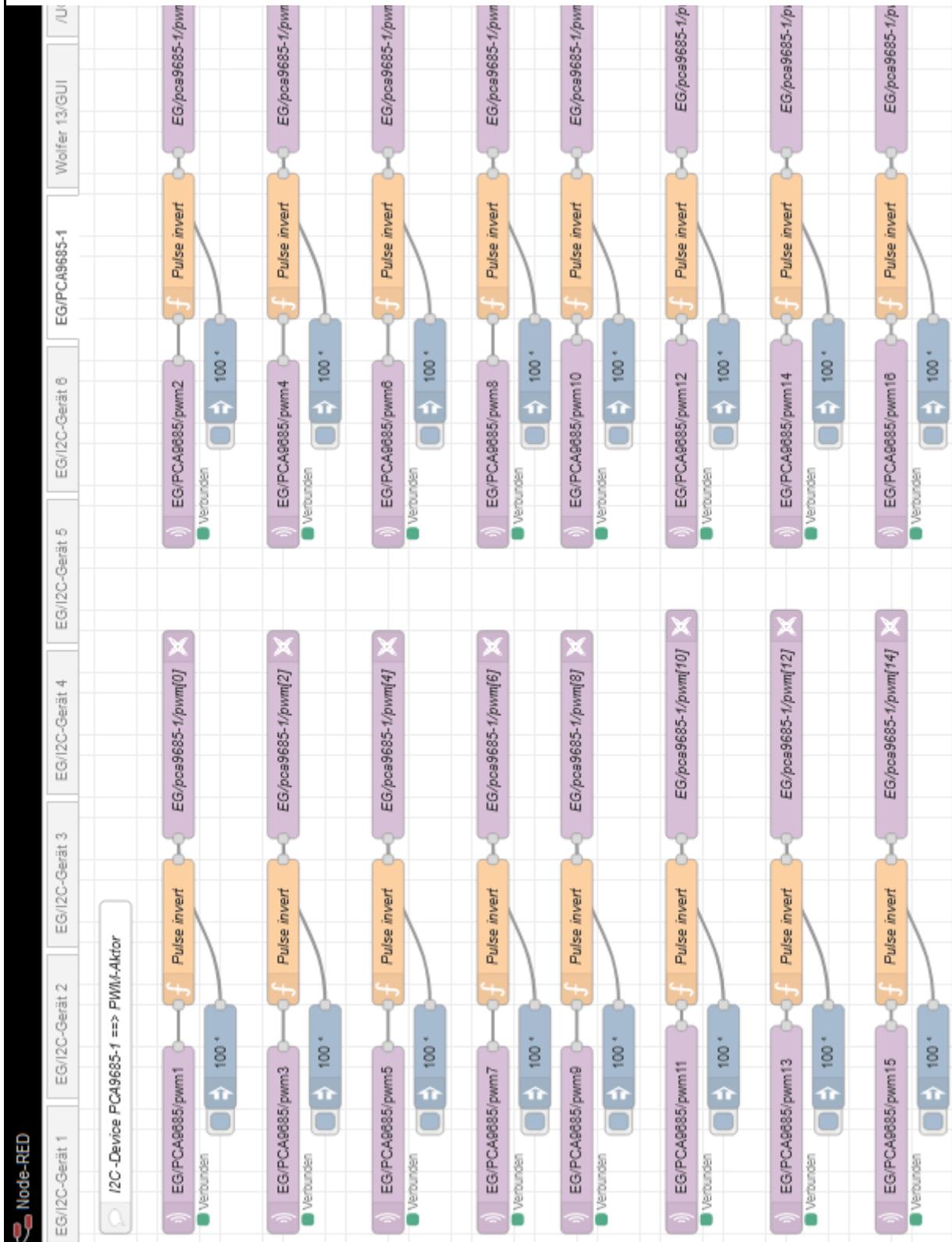
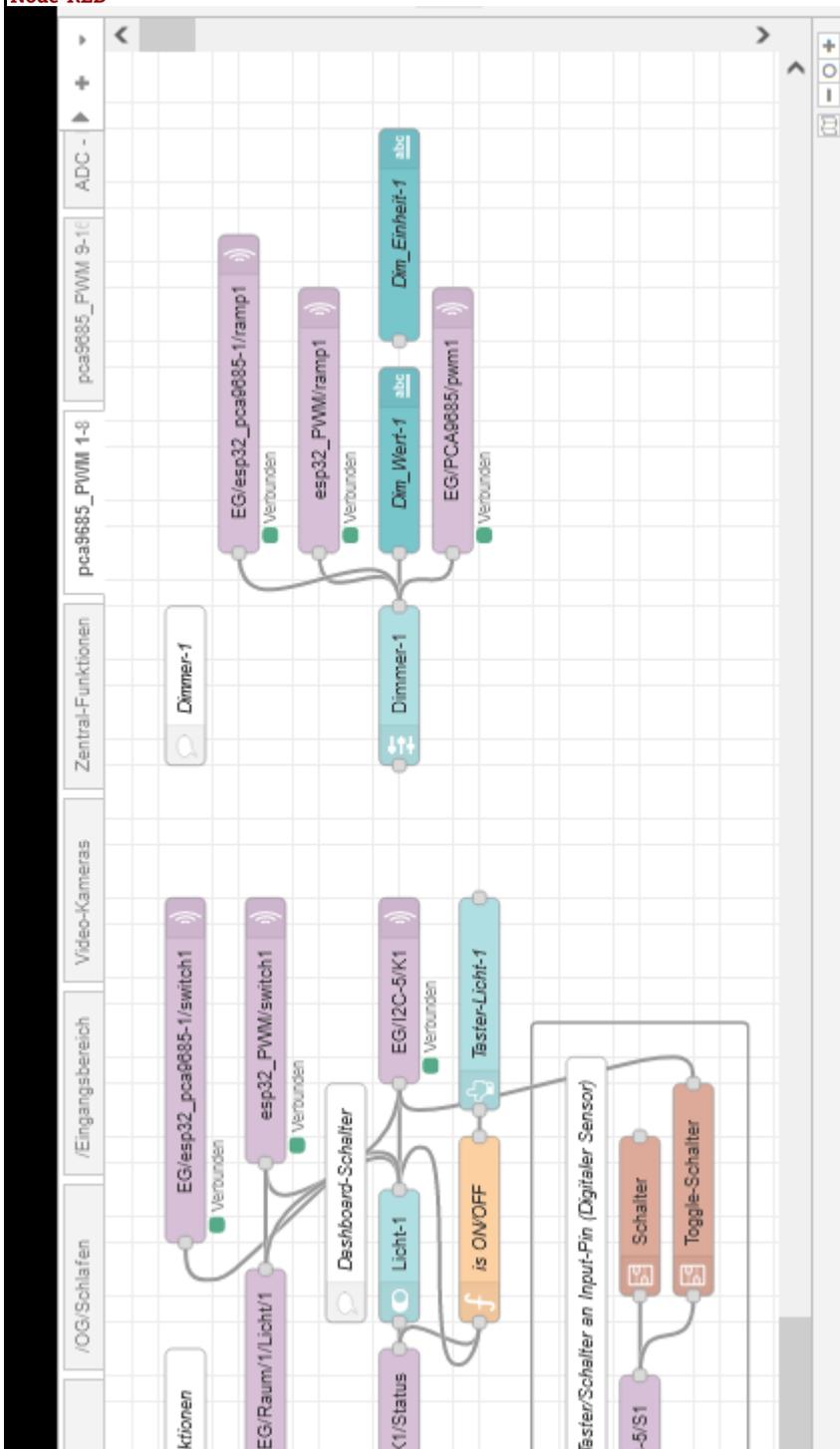


ABBILDUNG 6: Node-RED Application Flow Licht/Dimmer mit Schalter/ Raum Beispiel



3.0 Installation

In Abbildung 7 auf Seite 65 ist der Dimmer/Motion - Controller **DMC-32/TRIACC-8** (Bild oben) in Kombination mit PWM-Leistungstreiber und Triac-Leistungstreiber im Hutschinenengehäuse in der Installation in einem Schaltschrank Einbaufeld (Bild unten) dargestellt. In der vorliegenden Installation sind 4 Hutschienen-Module in einer



Dimmer/Motion-Controller

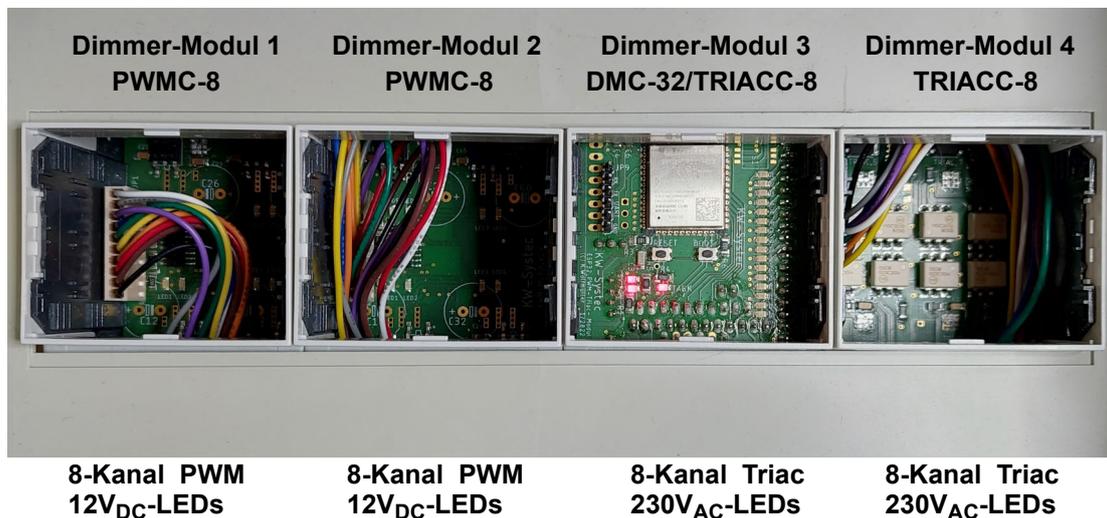


ABBILDUNG 7: Dimmer/Motion - Controller **DMC-32/TRIACC-8** mit integriertem Triac-Leistungstreiber Modul **TRIACC-8** (oben), **Beispiel Installation** (unten): Dimmer/Motion-Controller **DMC-32/TRIACC-8** mit 2x PWM-Leistungstreiber Modul **PWMC-8** und 1xTriac-Leistungstreiber Modul **TRIACC-8**

Reihe eines Schaltschrank Standardfeldes angeordnet. In dieser Konfiguration stehen insgesamt 16 Leistungsstufen (2 x 8 pro Modul) zur Dimmung von jeweils 12V_{DC} LEDs (oder Halogen-Lampen) mit PWM (PulsWeiten Modulation) zur Verfügung. Zusätzlich sind 16 Leistungsstufen (2 x 8 pro Modul) zur Dimmung von jeweils 230 V_{AC} LEDs (oder Glühlampen) über Phasenanschnitt Steuerung vorhanden. Die Triac-Leistungstreiber Module könnten auch durch PWM-Leistungstreiber Module ersetzt werden. Damit lassen sich sehr flexibel unterschiedliche Beleuchtungskonzepte bei unterschiedlichen Leuchtmitteln realisieren.

W *SmartHome* *Open*

Relais Modul Schließer





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

Relais Modul Schließer

1.0 Hardware Beschreibung

Funktion:	Schließer am Arbeitskontakt	
Bezeichnung:	RKS-8	
Anzahl Kontakte:	8	
Schaltkontakt A1.x/A2.x:	Steuerspannung	$V_{DD} = +24V_{DC}$
	Innenwiderstand	$R_i = 1440\Omega$
Arbeitskontakt Kx.1/Kx.2:	Spannung / Strom	$230V_{AC} / 16A$

TABELLE 1. Schaltzustand

Schaltkontakt A1.x / A2.x	Arbeitskontakt Kx.1 / Kx.2	x = 1 ... 8
OFF	OFF	
ON	ON	

Die Schaltkontakte können in Aktiv-HIGH-Logik mit P-Kanal MOSFET Leistungstreiber-Endstufen oder in Aktiv-LOW-Logik mit N-Kanal MOSFET Leistungstreiber-Endstufen betrieben werden. (Fußnoten in Abbildung 1, „Schaltplan Relais-Modul Schließer,“ auf Seite 70 beachten)

Die Ansteuerung der Relais K1 - K8 erfolgt über die Kontakte 3 - 10 des PSK-Steckers mit einem PSK-Kabel s. Abbildung 2, „PSK - Kabel (10-polig),“ auf Seite 71. Die Kontakte 1,2 sind alternativ gemäß gewählter Ansteuerlogik (s. Fußnote) zu betreiben.

Die Arbeitskontakt Belegung an den Klemmen ist in Abbildung 3 auf Seite 71 gegeben.

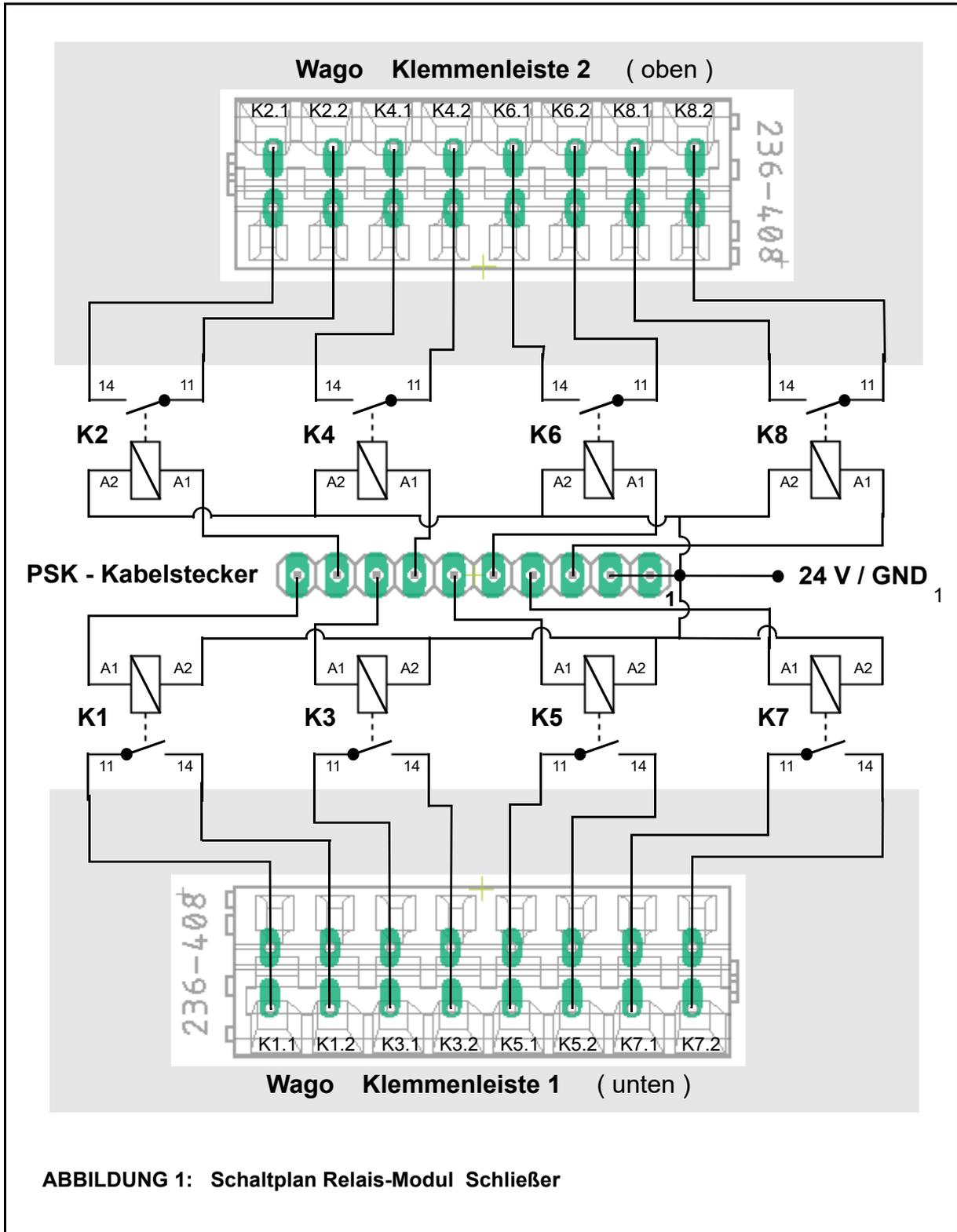
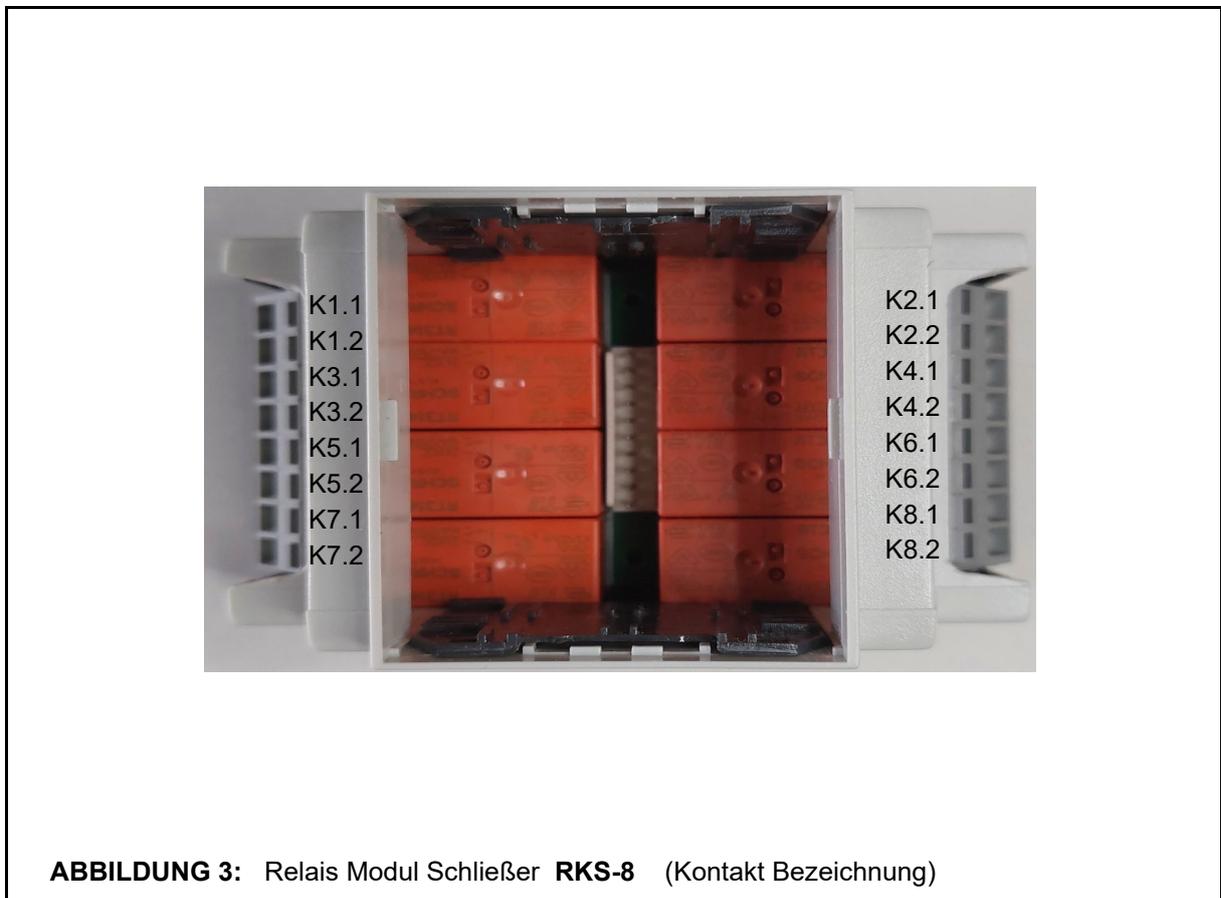
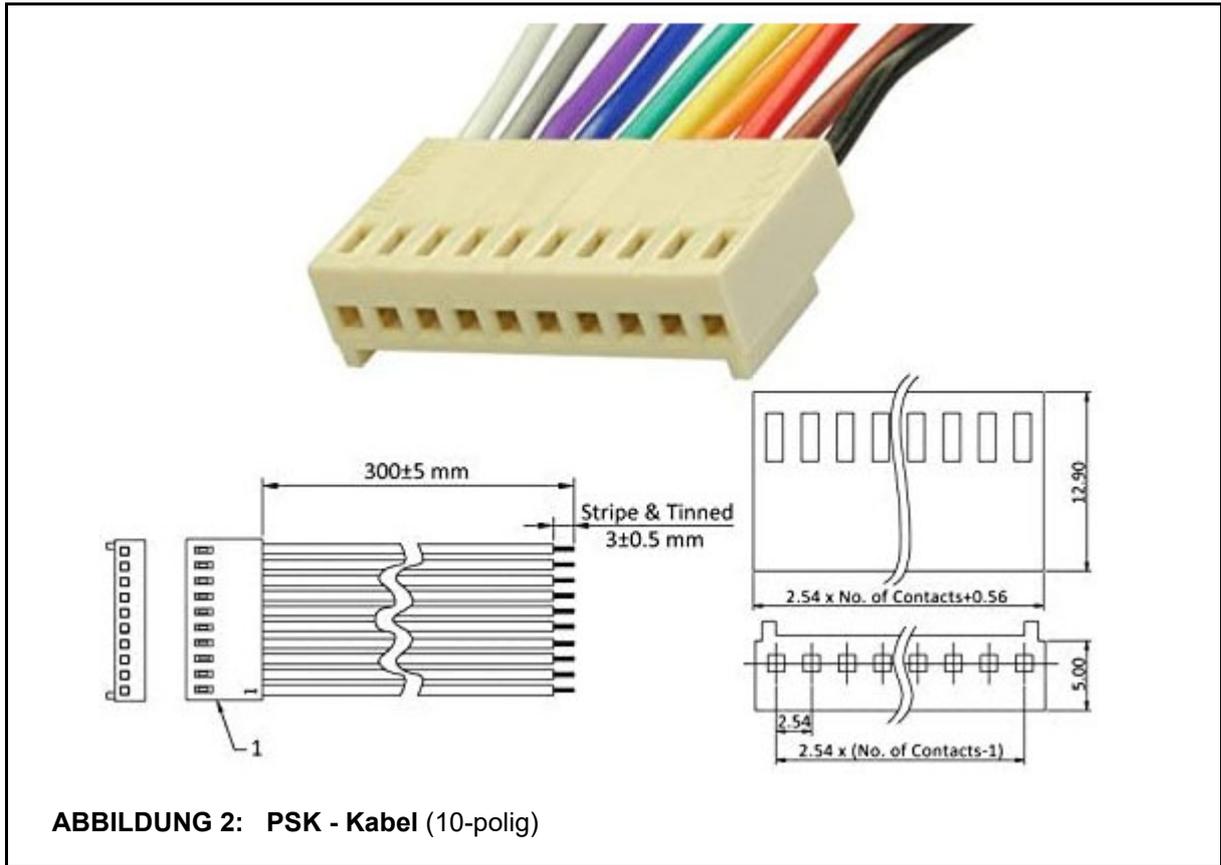


ABBILDUNG 1: Schaltplan Relais-Modul Schließer

¹ Fußzeilen

1. alternativ: $V_{DD} = +24V$ bei aktiv LOW-Logik der Leistungsendstufen an Schaltkontakten
 $V_{SS} = GND$ bei aktiv HIGH-Logik der Leistungsendstufen an Schaltkontakten



W *SmartHome* *Open*

Relais Modul Wechsler





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

Relais - Modul Wechsler

1.0 Hardware Beschreibung

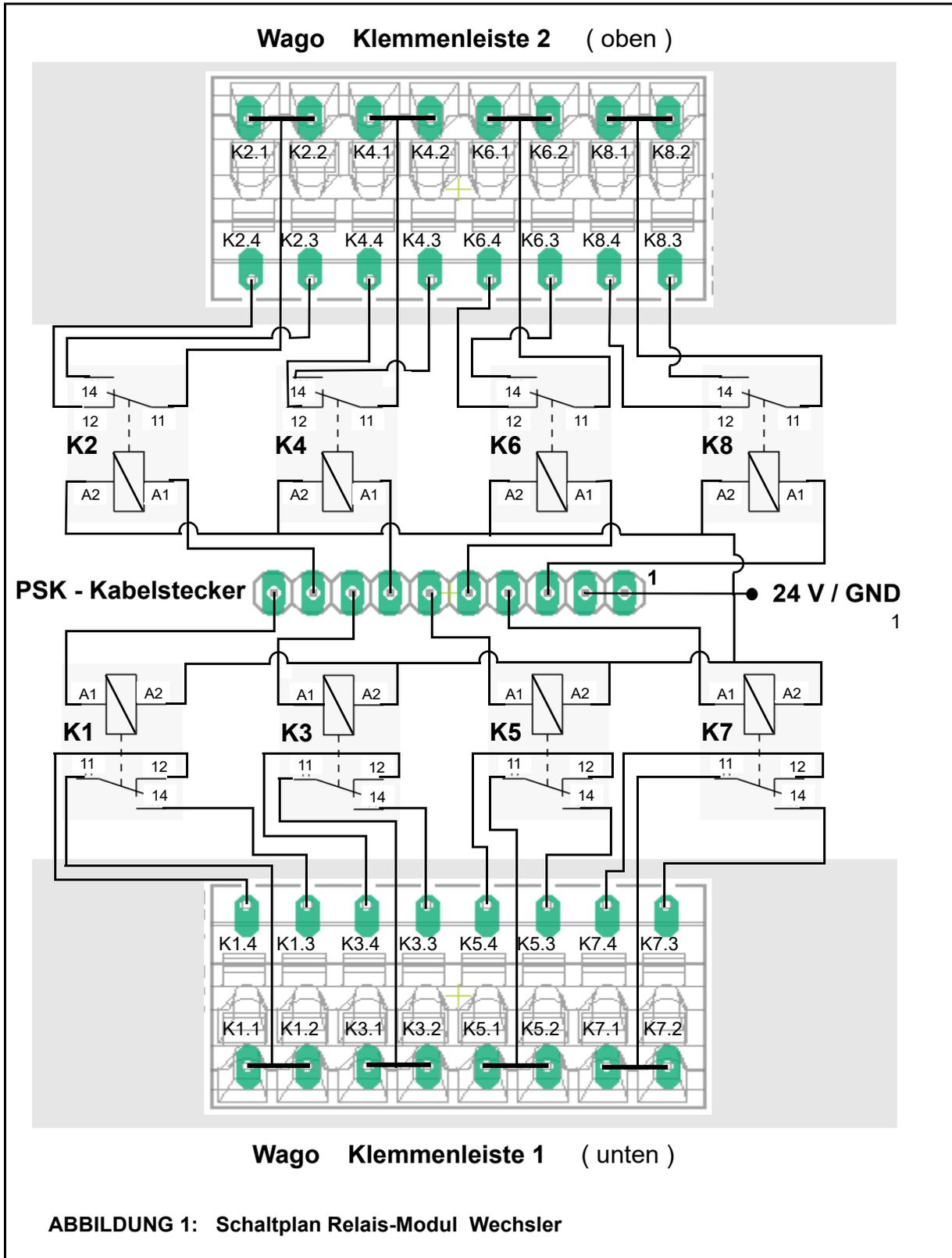
Funktion:	Wechsler an den Arbeitskontakten	
Bezeichnung:	RKW-8	
Anzahl Kontakte:	16	
Schaltkontakt A1.x/A2.x:	Steuerspannung	$V_{DD} = +24V_{DC}$
	Innenwiderstand	$R_i = 1440\Omega$
Arbeitskontakt Kx.1/Kx.2:	Spannung / Strom	$230V_{AC} / 16A$

TABELLE 1. Schaltzustand

Schaltkontakt A1.x / A2.x	Arbeitskontakt Kx.1 / Kx.3	Arbeitskontakt Kx.1 / Kx.4	x = 1 ... 8
OFF	OFF	ON	
ON	ON	OFF	

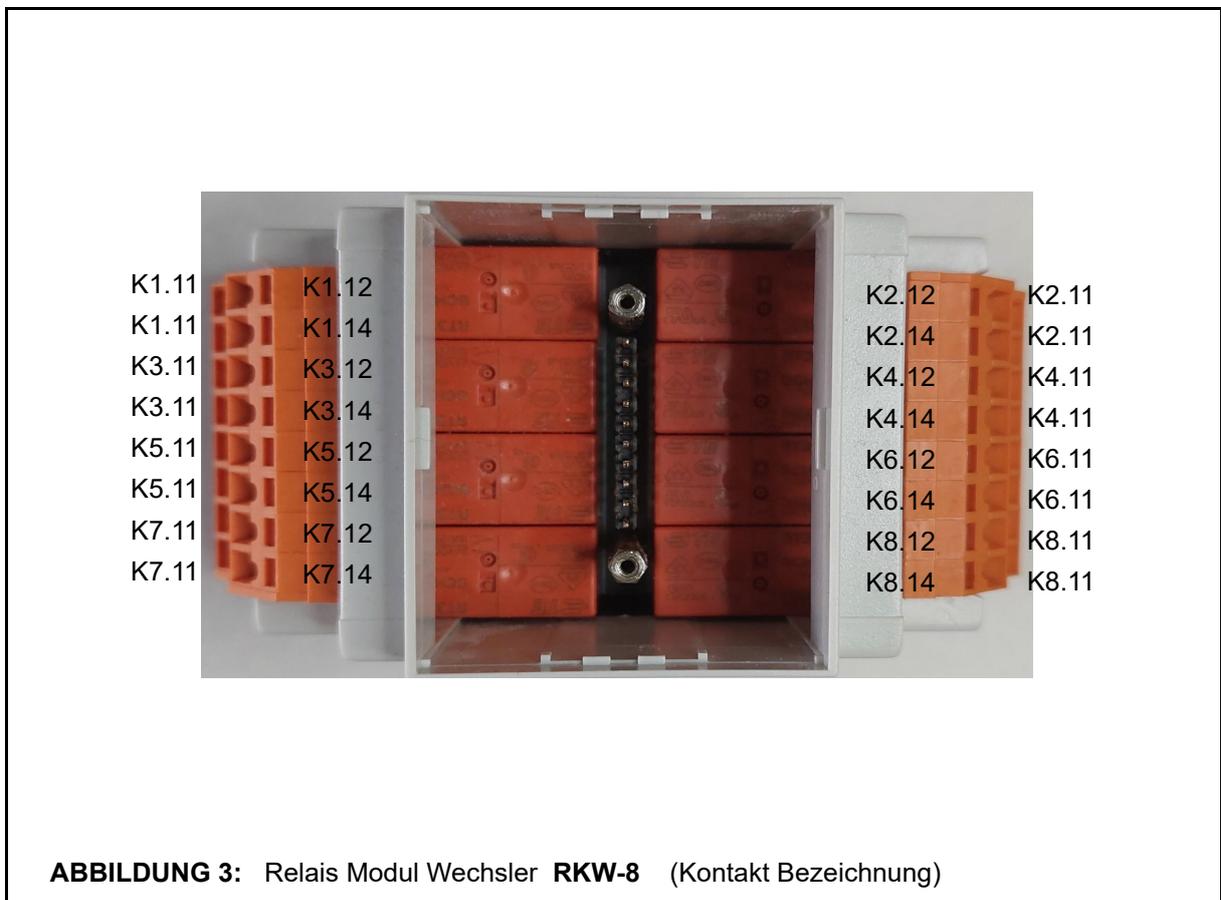
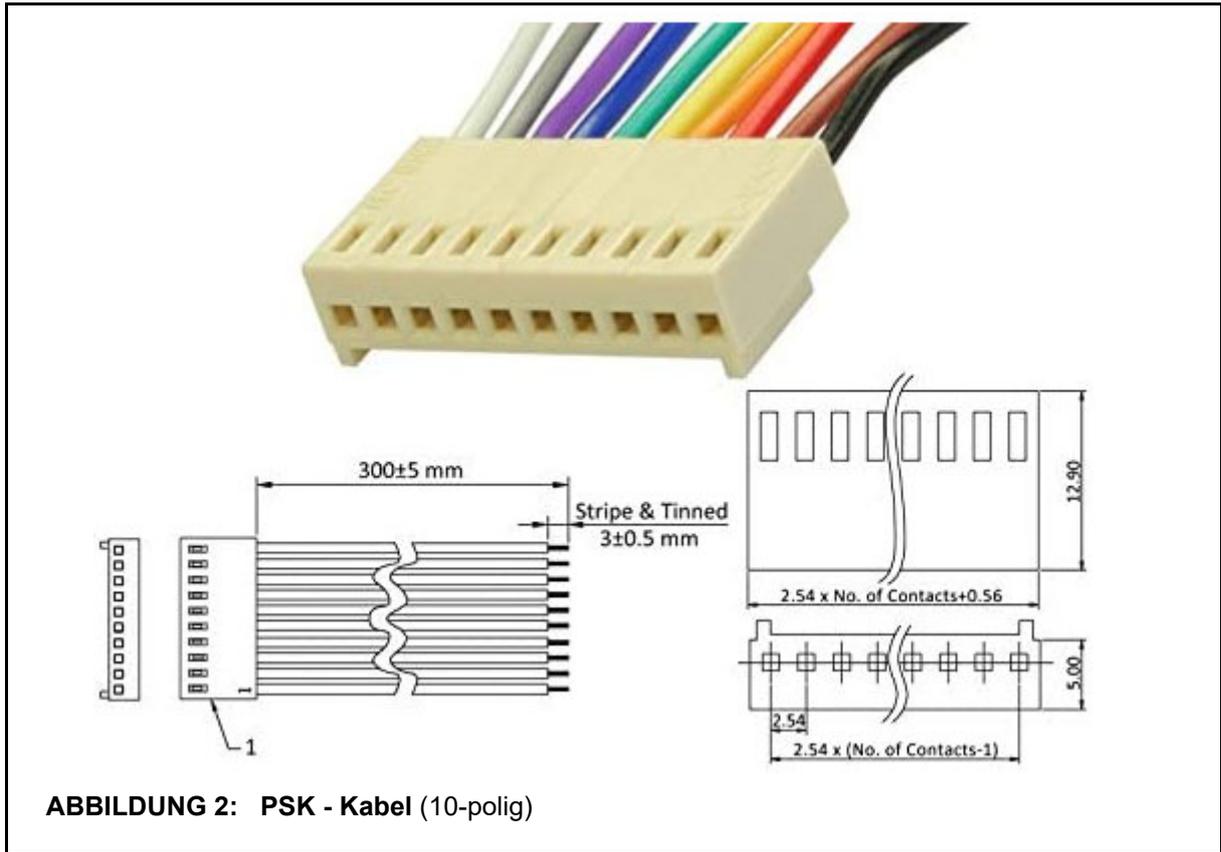
Die Schaltkontakte können in Aktiv-HIGH-Logik mit P-Kanal MOSFET Leistungstreiber-Endstufen oder in Aktiv-LOW-Logik mit N-Kanal MOSFET Leistungstreiber-Endstufen betrieben werden. (Fußnoten in Abbildung 1, „Schaltplan Relais-Modul Wechsler,“ auf Seite 76 beachten)

Die Ansteuerung der Relais K1 - K8 erfolgt über die Kontakte 3 - 10 des PSK-Steckers mit einem PSK-Kabel s. Abbildung 2, „PSK - Kabel (10-polig),“ auf Seite 77. Die Kontakte 1,2 sind alternativ gemäß gewählter Ansteuerlogik (s. Fußnote) zu betreiben.



¹ Fußzeilen

1. alternativ: $V_{DD} = +24V$ bei aktiv LOW-Logik der Leistungsendstufen an Schaltkontakten
 $V_{SS} = GND$ bei aktiv HIGH-Logik der Leistungsendstufen an Schaltkontakten



W *SmartHome* *Open*

Relais Modul Roll-Laden





KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

Relais Modul Roll-Laden

1.0 Hardware Beschreibung

Funktion:	Roll-Laden Ansteuerung an den Arbeitskontakten		
Bezeichnung:	RKR-4		
Anzahl der Roll-Läden:	4 (jeweils 2 Kontakte für „AUF“- und „AB“- Funktion)		
Schaltkontakt A1.x/A2.x:	Steuerspannung	$V_{DD} = +24V_{DC}$	
	Innenwiderstand	$R_i = 1440\Omega$	
Arbeitskontakt Kx.1/Kx.2:	Spannung / Strom	$230V_{AC} / 16A$	

Die Roll-Laden Kontakte AB und AUF sind für jeden Roll-Laden physikalisch verriegelt. Damit ist ausgeschlossen, dass an einem Roll-Laden die AB- und AUF-Leitungen gleichzeitig aktiv geschaltet sein können. (s. Tabelle 1 auf Seite 81, Tabelle 2 auf Seite 81, Tabelle 3 auf Seite 82 und Tabelle 4 auf Seite 82)

TABELLE 1. Schaltzustand Roll-Laden 1 (Relais K1, K2)

Schaltkontakt A1.1/ A2.1	Schaltkontakt A1.2 / A2.2	Arbeitskontakt K1.1 / AB-1	Arbeitskontakt K2.2 / AUF-1
OFF	OFF	OFF	OFF
ON	OFF	ON	OFF
OFF	ON	OFF	ON
ON	ON	OFF	OFF

TABELLE 2. Schaltzustand Roll-Laden 2 (Relais K3, K4)

Schaltkontakt A1.3/ A2.3	Schaltkontakt A1.4 / A2.4	Arbeitskontakt K3.1 / AB-2	Arbeitskontakt K4.2 / AUF-2
OFF	OFF	OFF	OFF
ON	OFF	ON	OFF
OFF	ON	OFF	ON
ON	ON	OFF	OFF

TABELLE 3. Schaltzustand Roll-Laden 3 (Relais K5, K6)

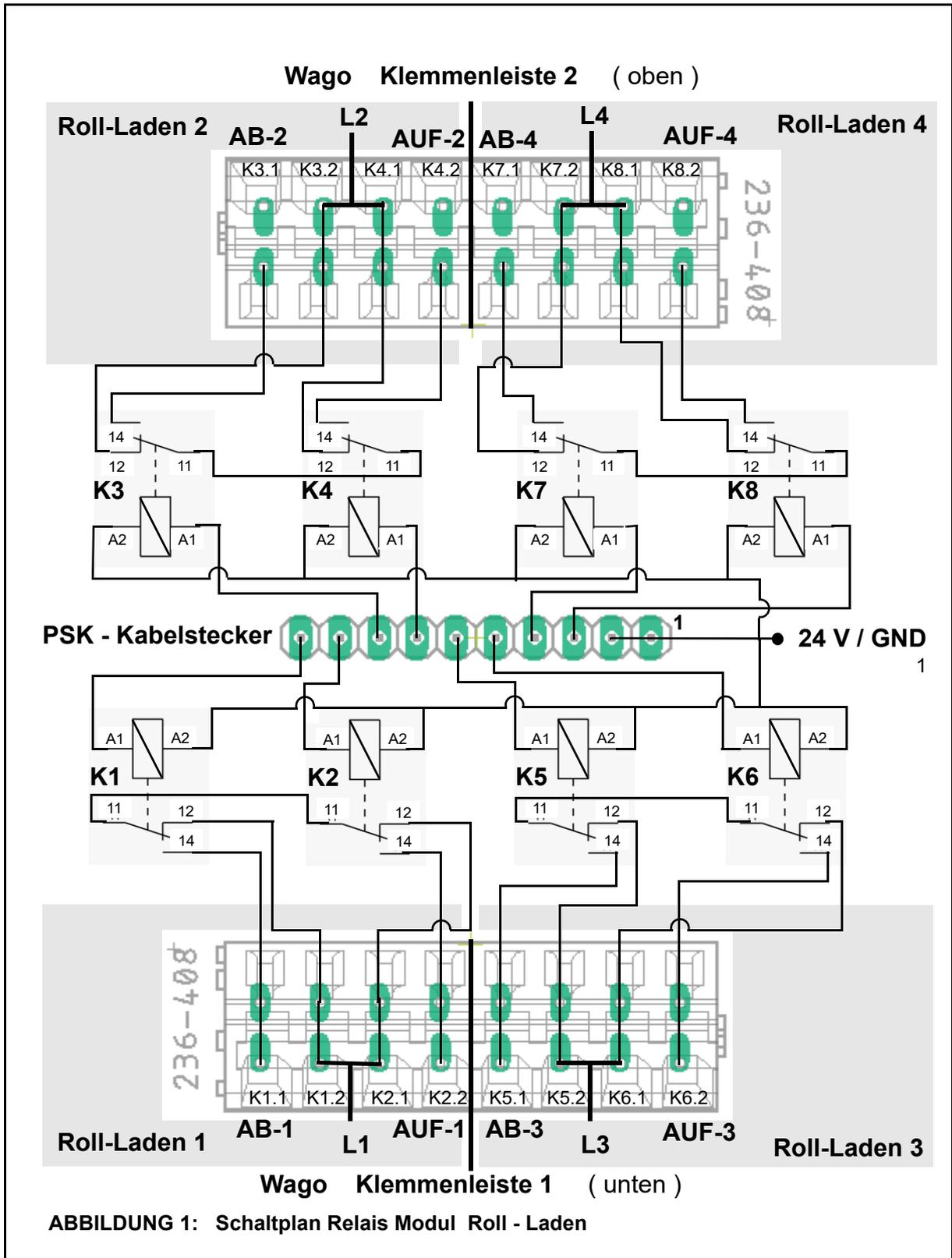
Schaltkontakt A1.5/ A2.5	Schaltkontakt A1.6 / A2.6	Arbeitskontakt K5.1 / AB-3	Arbeitskontakt K6.2 / AUF-3
OFF	OFF	OFF	OFF
ON	OFF	ON	OFF
OFF	ON	OFF	ON
ON	ON	OFF	OFF

TABELLE 4. Schaltzustand Roll-Laden 4 (Relais K7, K8)

Schaltkontakt A1.7/ A2.7	Schaltkontakt A1.8 / A2.8	Arbeitskontakt K7.1 / AB-4	Arbeitskontakt K8.2 / AUF-4
OFF	OFF	OFF	OFF
ON	OFF	ON	OFF
OFF	ON	OFF	ON
ON	ON	OFF	OFF

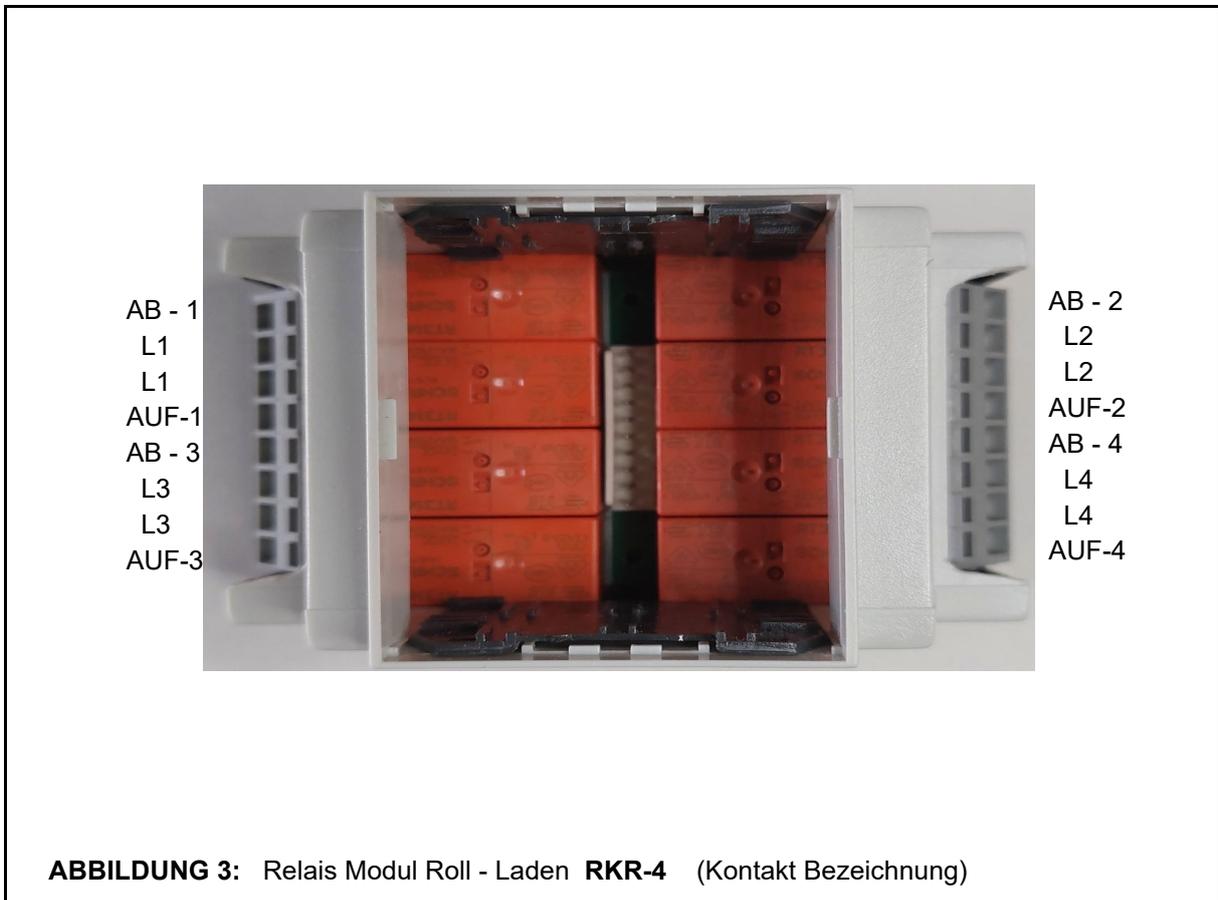
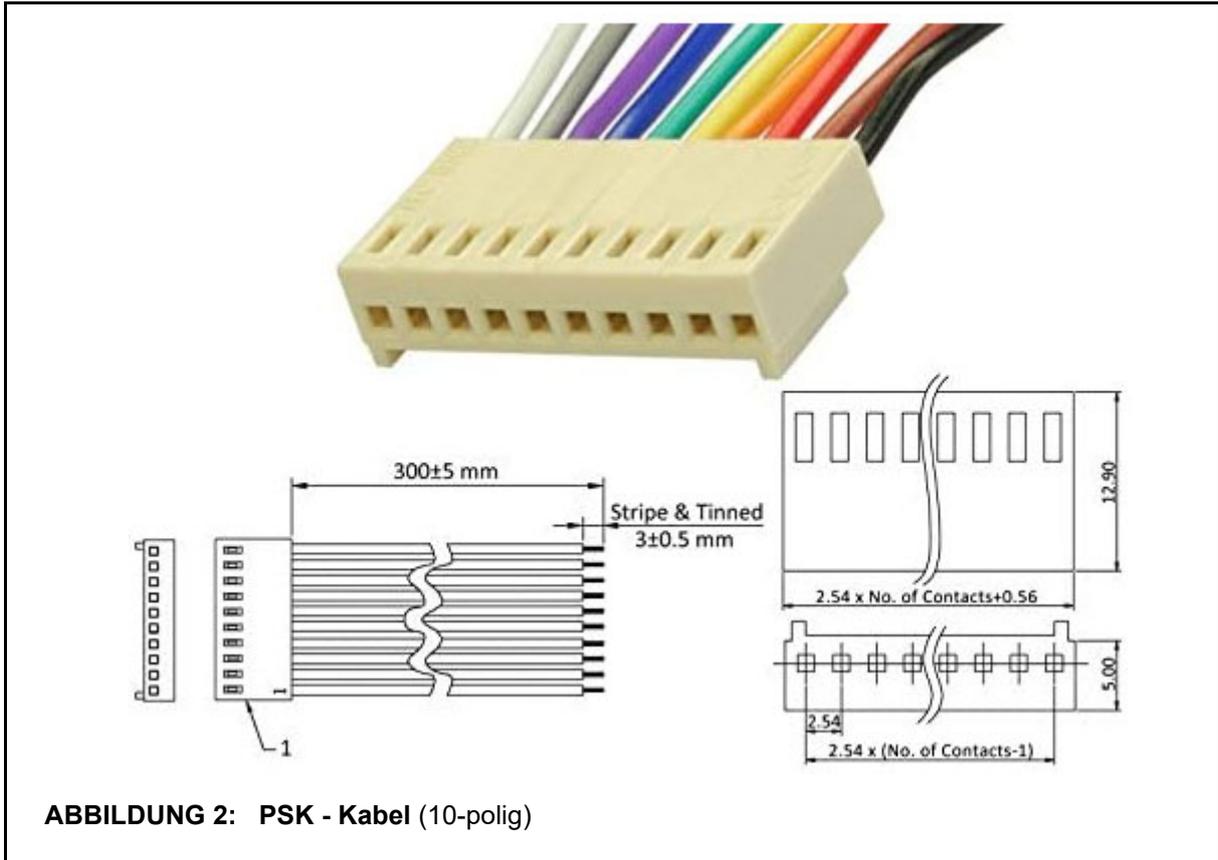
Die Schaltkontakte können in Aktiv-HIGH-Logik mit P-Kanal MOSFET Leistungstreiber-Endstufen oder in Aktiv-LOW-Logik mit N-Kanal MOSFET Leistungstreiber-Endstufen betrieben werden. (Fußnoten in Abbildung 1, „Schaltplan Relais Modul Roll - Laden,“ auf Seite 83 beachten)

Die Ansteuerung der Relais K1 - K8 erfolgt über die Kontakte 3 - 10 des PSK-Steckers mit einem PSK-Kabel s. Abbildung 2, „PSK - Kabel (10-polig),“ auf Seite 84. Die Kontakte 1,2 sind alternativ gemäß gewählter Ansteuerlogik (s. Fußnote) zu betreiben.



¹ Fußzeilen

1. alternativ: $V_{DD} = +24V$ bei aktiv LOW-Logik der Leistungsendstufen an Schaltkontakten
 $V_{SS} = GND$ bei aktiv HIGH-Logik der Leistungsendstufen an Schaltkontakten



KW SmartHome Open

KW-SmartHome Referenzprojekt „ Wolfer - 13 “

Raspberry Pi OS (64-bit)
A port of Debian Bullseye
Veröffentlicht: 2022-04-01

ESPRESSIF

mosquitto Mosquitto broker MQTT

Node-RED

SmartHome Open

System
CPU - Temperatur
69.1 °C



KW - Systemtec

Karlheinz Wolfmüller - Systemtechnik

Wolferstraße 13

D - 75031 Eppingen

<http://www.kwsystemtec.de>

KW-Smarthome Referenzprojekt „ Wolfer - 13 “

Das Gebäude des Referenzprojekts ist ein Einfamilienhaus mit 3 Etagen EG, OG und UG. In Abbildung 1 auf Seite 87 ist ein Foto der für die Schaltschrank Installation

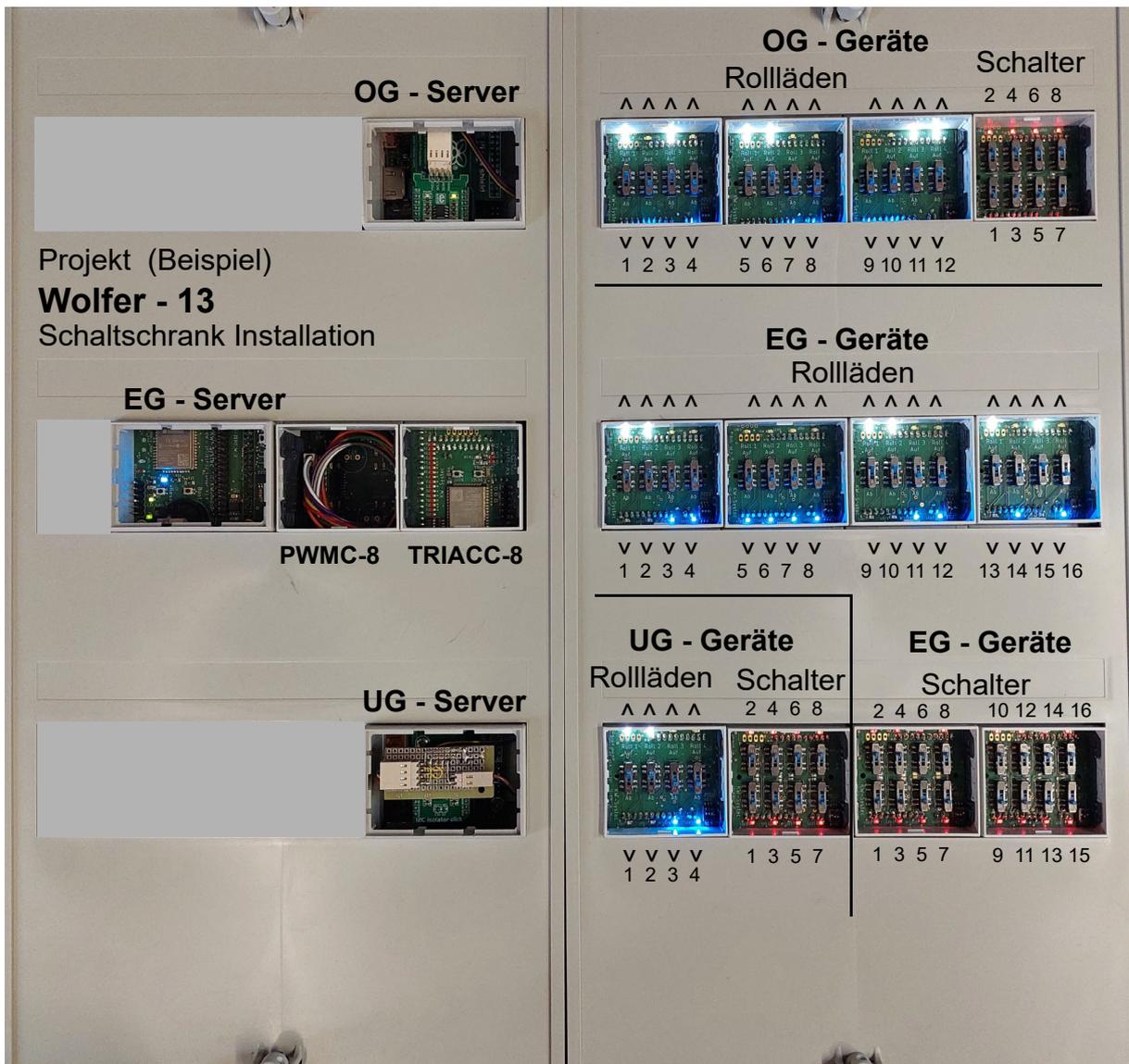
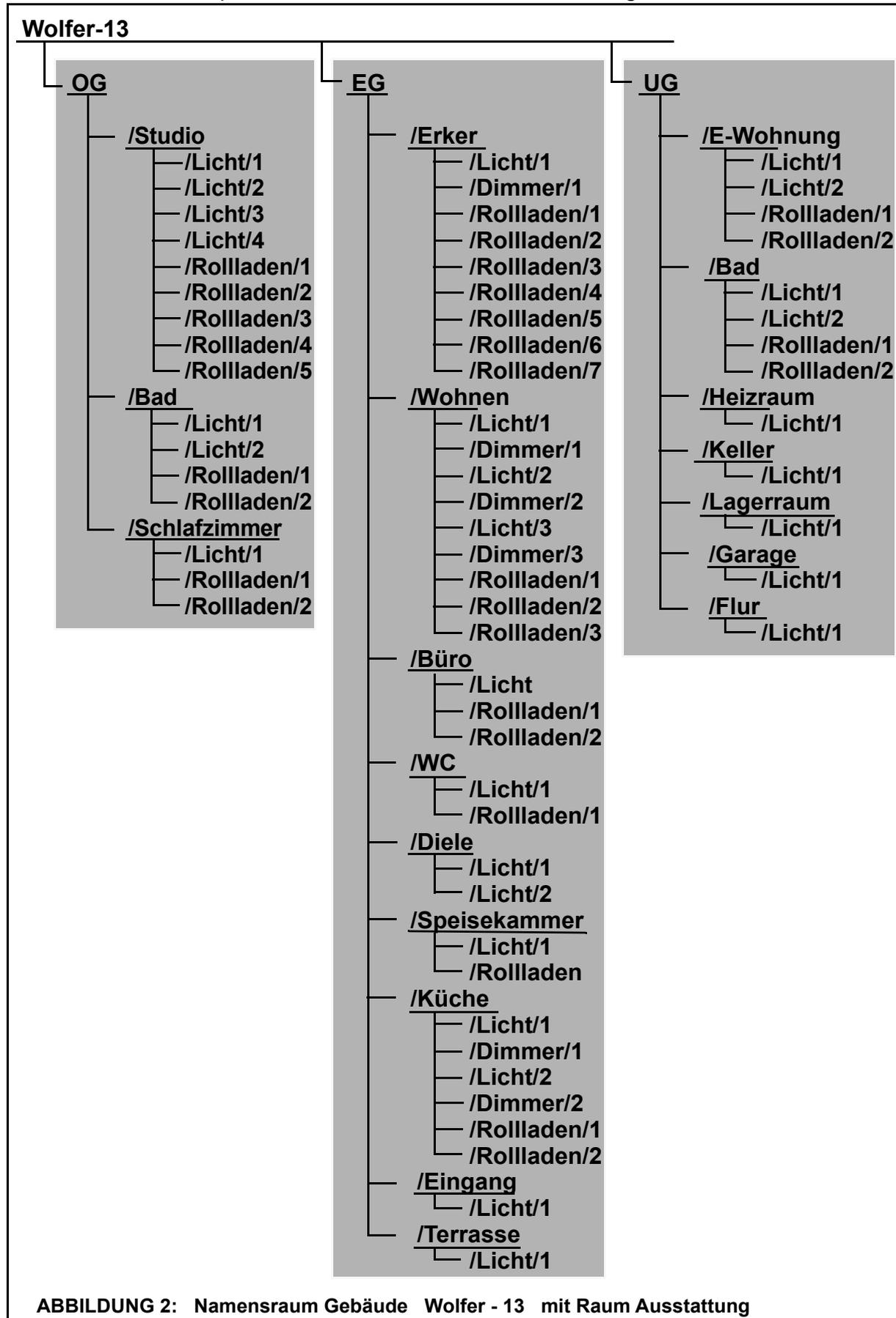


ABBILDUNG 1: Schaltschrank Felder des realisierten Projektes im Testbetrieb

benötigten Felder des realisierten Projekts zu sehen. Aus den Plänen ergibt sich der Namensraum des Gebäudes mit den Raumausstattungen gemäß Abbildung 2 auf Seite 88 . Mit dieser Baumstruktur lässt sich jedes Sensor/Aktor-Element mit den

entsprechenden Datenpunkt Variablen in den **Raum Applikationen Flows** mit den exakt identischen Topic Namen im MQTT-Protokoll eindeutig adressieren.



Die verwendeten Topic Namen können also direkt aus der Baumstruktur im Gebäude Namensraum übernommen werden. In der gleichen Logik werden in den **Geräte Treiber Flows** die Topic Namen der Geräte Datenpunkt Variablen festgelegt. Es können damit die Sensor/Aktor-Bindungen eindeutig hergestellt werden. Die

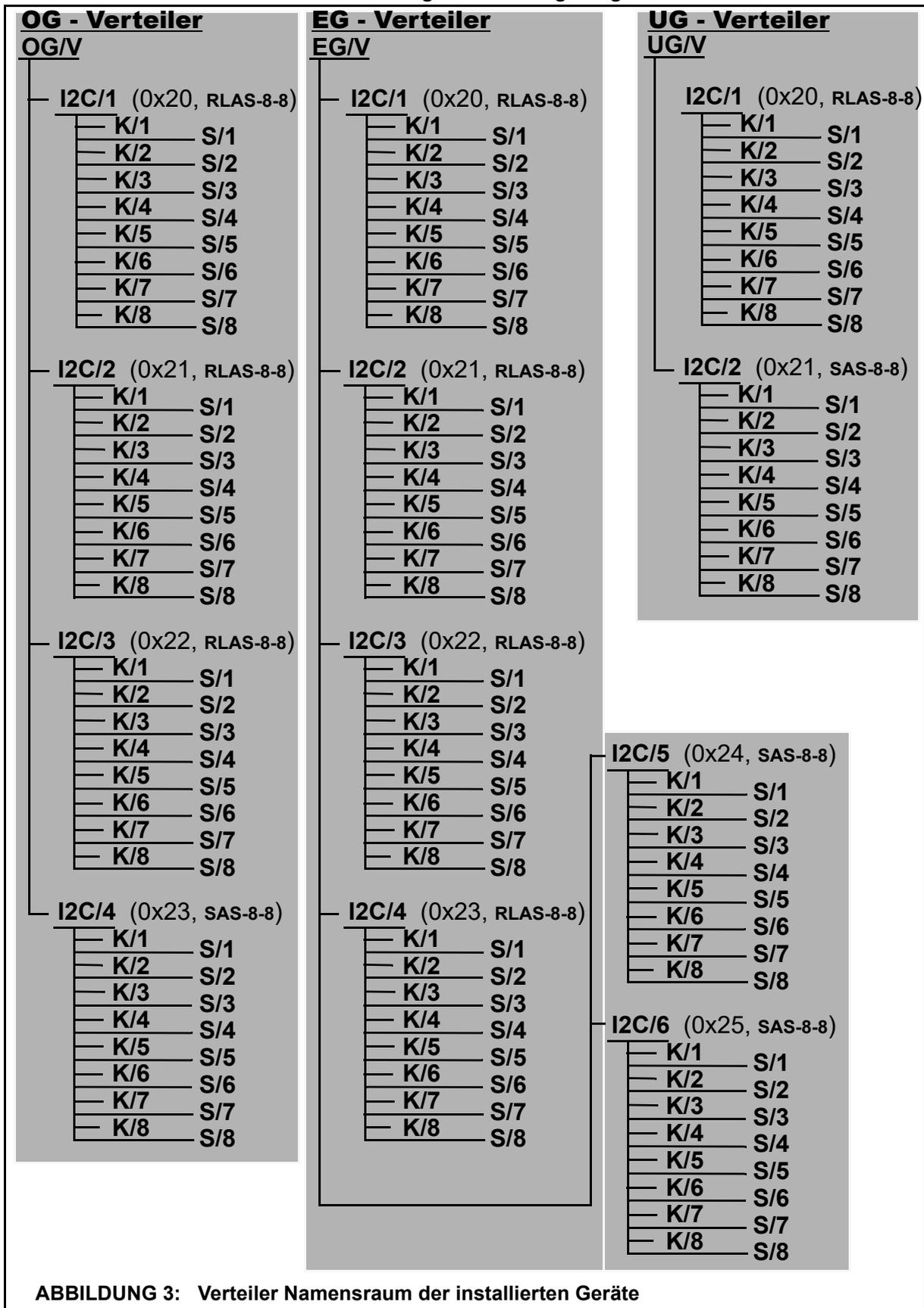


ABBILDUNG 3: Verteiler Namensraum der installierten Geräte

allgemeinen numerierten Namen Bezeichnungen gemäß Abbildung 3 auf Seite 9 können in einem konkreten Projekt durch die echten Namen der Räume gemäß der Bezeichnungen in den Plänen ersetzt werden. Dadurch werden die Programm Flows sehr viel anschaulicher und können als Dokumentation nachvollziehbar verwendet werden.

Es werden die folgenden Geräte für die einzelnen Etagen gemäß Abbildung 3 auf Seite 89 benötigt.

OG:

- 3 x Rollladen-Modul **RLAS-8-8** (12 Rollläden)
- 1 x Schalter-Modul **SAS-8-8** (8 Schalt-Aktoren)

EG:

- 4 x Rollladen-Modul **RLAS-8-8** (16 Rollläden)
- 2 x Schalter-Modul **SAS-8-8** (16 Schalt-Aktoren)
- 1 x PWM-Modul **PWMC-8** (8 Dimmer Aktoren ==> 12V/DC - LED)
- 1 x Triac-Modul **TRIACC-8** (8 Dimmer Aktoren ==> 230V/AC - LED)

UG:

- 1 x Rollladen-Modul **RLAS-8-8** (4 Rollläden)
- 1 x Schalter-Modul **SAS-8-8** (8 Schalt-Aktoren)

Daraus ergibt sich eine installierte Gesamtkapazität für die physikalisch installierten Sensoren und Aktoren:

- **32 x Rolläden** (64 Relais),
- **32 x Schalt - Aktoren** (32 Relais),
- **64 x Schalter/Taster Sensoren** (64 digital Eingänge),
- **8 x TRIAC** - Leistungsendstufen für 8 Licht-Dimmer 230V/AC- LED,
- **8 x PWM** - Leistungsendstufen für 8 Licht-Dimmer 12V/DC- LED.

Die Geräte der einzelnen Etagen sind in den jeweiligen Etagen-Verteiler montiert. Im OG- und UG-Verteiler kommt jeweils ein Low Power Server mit PI_Zero 2 W Prozessor Modulen zum Einsatz. Der High Performance Server mit den MQTT-Broker ist im EG-Verteiler installiert. Es werden 2 PI-4 Prozessor Module gemäß „2PI4/ESP32 - Server“ auf Seite 27 eingesetzt.

Alle physikalischen Taster/Schalter Sensoren sowie die Dimmer Sensoren sind alle auch als virtuelle grafische Objekte aus der Dashboard Node-RED Library in den Raum Tabs der grafische Bedienoberflächen realisiert. Damit wird konsequent auch die Bedienung der Raum Funktionen mit Touchscreen Geräten möglich. Damit ist der Trend im SmartHome Bereich zu einem zusätzliche Bedienkomfort z.B. über Smartphones konsequent unterstützt. In Abbildung 4 auf Seite 91 ist als Beispiel für die virtuellen Bedienfunktion das Raum Auswahl Menü Tab dargestellt, Für alle Räume existiert ein entsprechendes Raum Tab, das bei Aktivierung die virtuellen

Raumfunktionen selektiert. In Abbildung 5 auf Seite 92 ist das Raum Tab Wohnen mit

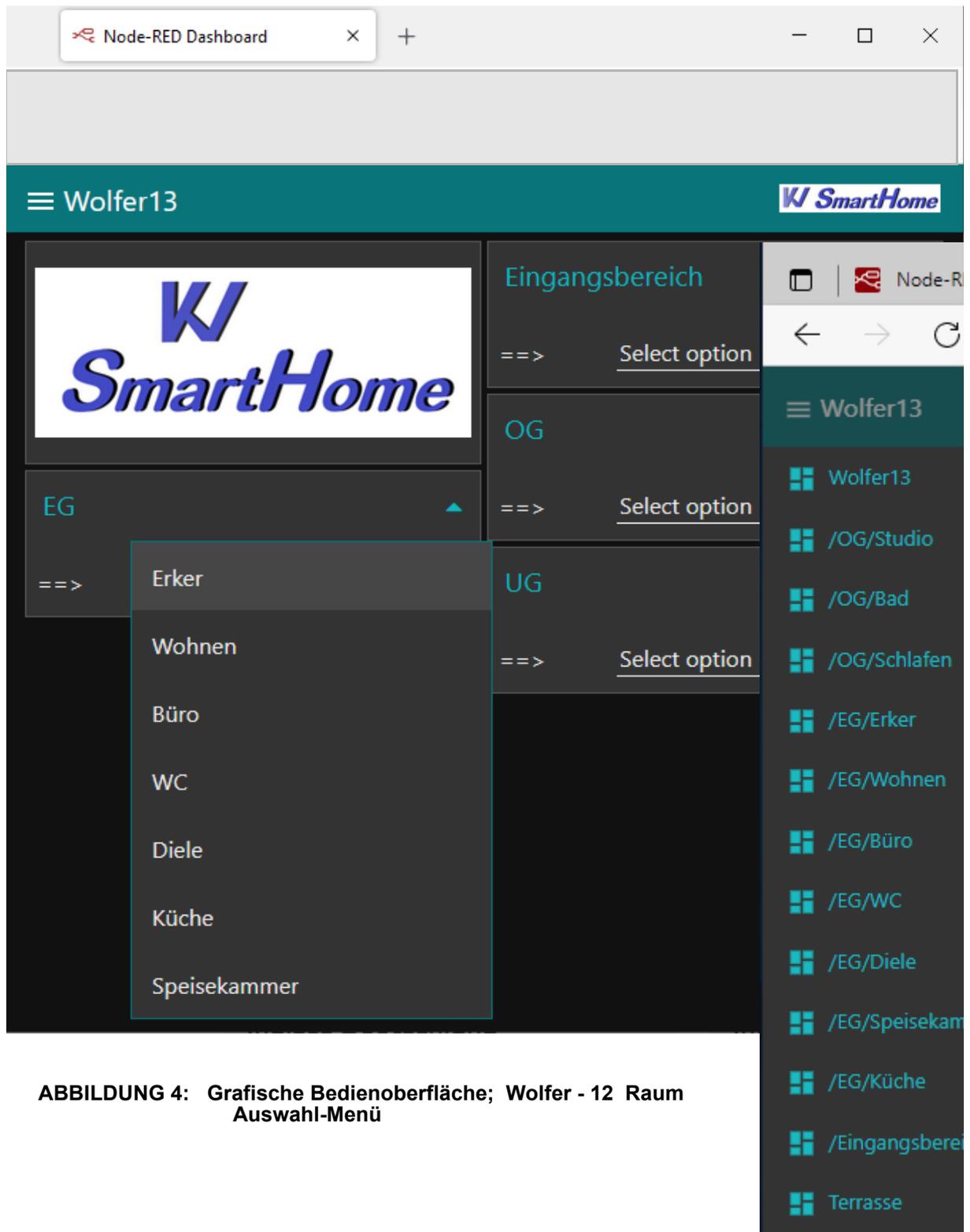


ABBILDUNG 4: Grafische Bedienoberfläche; Wolfer - 12 Raum Auswahl-Menü

den Licht- und Dimmer- Funktionen sowie der Roll-Laden Bedienung als exemplarisches Dashboard Beispiel dargestellt.

☰
Zentral-
Funktionen

Zeitschaltungen

Datum 23.06.2022

Uhrzeit 16:15

Sonnen - Aufgang/Untergang

Wohnen

Licht (Couch) AN

Dimmer 100 %

Licht (Tisch) AUS

Dimmer 0 %

Licht (Kamin) AN

Dimmer 68 %

Rollladen 1, Schiebetüre

Rollladen 2, Nord

Zentrale Licht-Steuerung

Alle Lichter ==> AN / AUS

Zentral-Schalter AUS

OG - Lichter ==> AN / AUS

OG Zentral-Lichtschalter AN

EG - Lichter ==> AN / Aus

EG Zentral-Lichtschalter AUS

UG - Lichter ==> AN / AUS

UG Zentral-Lichtschalter AUS

ABBILDUNG 5: Raum Tab „Wohnen“
Tab „Zentrale
Lichtsteuerung“