

spaceLYnk

Benutzerhandbuch



Version	Beschreibung	Erstellt
A	Erste Version	2.6.2014
B	Version für Firmware 1.1	1.3.2015

Inhalte

1	Produktsicherheit	10
1.1	Konnektivität	10
1.2	Sicherheitsfunktionen	10
1.3	Bedienung.....	10
1.4	Wartung.....	10
1.5	Patchverwaltung.....	11
1.6	Auf Werkseinstellungen zurücksetzen	11
1.7	Einschränkungen	11
2	Kurzanleitung.....	12
2.1	Startseite	13
2.2	Standardkonfiguration	14
2.3	Erkennen der spaceLYnk IP-Adresse	16
2.4	Firmware-Upgrade	18
2.5	spaceLYnk für KNX/EIB Netzwerk-Konfigurationsverwaltung mit ETS3	19
2.6	spaceLYnk für KNX/EIB Netzwerk-Konfigurationsverwaltung mit ETS4.....	20
2.7	spaceLYnk für KNX/EIB Netzwerk-Konfigurationsverwaltung mit ETS5.....	21
2.8	KNX und IP-Router-Einstellungen.....	22
2.9	Visualisierung erstellen für iPad/PC	23
2.9.1	Objekte importieren.....	23
2.9.2	Erstellen Sie eine Gebäude-/Etagen-Struktur und fügen Sie der Karte Objekte hinzu .	24
2.9.3	Der Neu erstellten Visualisierungskarte Objekte hinzufügen	26
2.9.4	Starten der Visualisierung auf einem Smartphone (in diesem Fall iPod).....	27
2.9.5	Starten der Visualisierung auf einem PC, Tablet oder jeglichem anderen Touch-Gerät mit großem Bildschirm.....	27
3	Erweiterte Anleitung	28
3.1	Hauptseite Konfigurator.....	28
3.2	Dienstprogramme	30
3.2.1	Importieren *.ESF-Datei	30
3.2.2	Nachbarn importieren.....	30
3.2.3	Zurücksetzen / Aufräumen.....	31
3.2.4	Auf Werkseinstellungen zurücksetzen	31
3.2.5	Datum und Zeit.....	32
3.2.6	Updates installieren.....	32

3.2.7	Sicherung	32
3.2.8	Wiederherstellen.....	33
3.2.9	Allgemeine Konfiguration.....	33
3.2.10	Vis.configuration	34
3.3	System	35
3.3.1	System → System.....	35
3.3.2	System → Netzwerk	37
3.3.3	System → Dienste	44
3.3.4	System → Status	45
3.4	Objekte	48
3.4.1	Objektparameter	49
3.4.2	Ereignisskript	49
3.4.3	Objektwert festlegen.....	50
3.4.4	Parameter zur Objektvisualisierung	50
3.4.5	Benutzerdefinierter Textwert.....	52
3.4.6	Objekt-Steuerleiste.....	52
3.4.7	Objektfilter	53
3.4.8	Massenbearbeitung.....	53
3.5	Objektprotokolle	54
3.6	Terminplaner (Schedulers)	55
3.6.1	Neuen Termin hinzufügen.....	55
3.6.2	Termin Ereignisse	56
3.6.3	Feiertage.....	56
3.7	Trend-Protokolle	57
3.7.1	Neues Trendprotokoll hinzufügen.....	59
3.8	Vis. Struktur	60
3.8.1	Ebenen.....	60
3.8.2	Zweite Ebene	60
3.8.3	Plan.....	61
3.8.4	Layout	62
3.8.5	Widget	64
3.8.6	Beispiel einer Visualisierungsstruktur	65
3.8.7	Plan.....	65
3.8.8	Layout	66

3.8.9	Widget	66
3.8.10	Sortierung der Visualisierungsobjekte	67
3.9	Visualisierung	68
3.9.1	Struktur.....	69
3.9.2	Visualisierungskarte	70
3.9.3	Plan-Editor	70
3.9.4	Objekt	71
3.9.5	Verknüpfung	74
3.9.6	Kamera.....	75
3.9.7	Diagramme	76
3.9.8	Beschriftung.....	77
3.9.9	Bild.....	77
3.9.10	Messgerät.....	78
3.9.11	Rahmen.....	79
3.10	Vis. graphics.....	80
3.11	Benutzerzugriff	81
3.12	Scripting	83
3.12.1	Ereignisbasiert	84
3.12.2	Resident	85
3.12.3	Geplant	85
3.12.4	Benutzerbibliotheken	86
3.12.5	Allgemeine Funktionen.....	86
3.12.6	Start-Skript.....	87
3.12.7	Tools	87
3.12.8	Allgemeine Scripting-Beschreibung.....	88
3.12.9	Skript-Editor.....	89
3.13	Warnungen	91
3.14	Protokolle	92
3.15	Fehlerprotokoll	93
3.16	Hilfe	93
4	Modbus RTU und Modbus TCP.....	93
4.1	Charakteristiken	93
4.3	Modbus-Geräteprofile.....	95
4.3.1	Neue Profildefinition	96

4.4	Modbus RTU-Schnittstelle.....	98
4.4.1	Charakteristiken der Modbus RTU-Schnittstelle	98
4.4.2	RS 485 Bus Topologie-Merkmale.....	98
4.4.3	Konfigurationsbefehle Modbus RTU	101
4.5	Modbus TCP Interface	104
4.5.1	Charakteristiken der Modbus TCP-Schnittstelle.....	104
4.5.2	Konfigurationsbefehle Modbus TCP.....	104
4.6	Modbus-Funktionscodes und entsprechende Master-Funktionen.....	105
4.7	Modbus Masterfunktionen	109
4.8	Modbus Slavefunktionen	110
4.8.1	Empfangsdaten vom Master	110
4.8.2	Modbus-Zuordnung des Slave-Geräts festlegen	110
4.8.3	Handle-Slave.....	110
4.8.4	Get-Funktionen.....	110
4.8.5	Set-Funktionen	110
4.8.6	Rückruffunktionen.....	111
5	Port-Weiterleitung	112
5.1	Einführung	112
5.2	HTTP und HTTPS	112
5.2.1	HTTP.....	112
5.2.2	HTTPS.....	112
6	BACnet.....	113
6.1	Charakteristiken	113
6.2	spaceLYnk-Konfiguration.....	113
6.3	Objektexport	113
6.4	BACnet-Konfiguration.....	114
6.4.1	Konfiguration.....	114
6.4.2	BACnet-Objekte	114
6.5	BACnet Standardisiertes Geräteprofil	115
6.5.1	Liste aller unterstützten BACnet Interoperability Building Blocks (BIBBs).....	115
6.5.2	Unterstützte BACnet Objekttypen	115
6.5.3	Layer-Optionen Datenverbindung.....	115
6.6	Building Operation Workstation	116
7	RS-232 Serielle Leitung	117

7.1	Charakteristiken	117
7.2	Konfigurationsbefehle	117
8	USB 2.0	119
8.1	Charakteristiken	119
8.2	Konfigurationsbefehle	119
8.3	Durch Anbringen eines USB GSM-Adapters können SMS-Nachrichten gesendet und empfangen werden.	119
8.3.1	Befehlssyntax:.....	120
9	Blockprogrammierung.....	121
9.1	Einführung	121
9.2	Blockprogrammierung-Schnittstelle.....	121
9.3	Verwendungsbeispiele	123
9.3.1	AND von zwei KNX-Objekten.....	123
9.3.2	Eine Warnung wird erstellt, wenn der IF Objektwert das Maximum überschreitet... <td>123</td>	123
9.4	Benutzerdefinierte Funktionsblöcke	124
9.5	Spezielle Funktionsblöcke	126
9.5.1	Allgemein.....	126
9.5.2	Convertors	126
9.5.3	Logical.....	126
9.5.4	Statistical	126
10	LUA – Programmiersprache	127
10.1	Objektfunktionen	127
10.2	Gruppenkommunikation-Funktionen.....	128
10.3	Funktionen zurückgegebener Objekte	128
10.4	Datentyp-Funktionen	129
10.5	Datentypen.....	129
10.6	Funktionen Datenspeicherung	130
10.7	Warnfunktionen	131
10.8	Protokoll-Funktionen.....	132
10.9	Zeit-Funktionen	132
10.10	String-Funktionen	132
10.11	Eingabe- und Ausgabefunktionen	139
10.12	Funktion Skriptsteuerung	139
10.13	Konvertierungen.....	139
10.14	Bit-Operatoren	140

10.15	Eingabe- und Ausgabeeinrichtungen	140
10.16	Mathematische Funktionen	142
10.17	Tabellenbearbeitung	144
10.18	Betriebssystem-Einrichtungen	145
10.19	Erweiterte Funktions bibliothek	146
11	Skript beispiele	148
11.1	Binär filter	148
11.2	Binär-Gate mit Bit-Gate	148
11.3	Gate mit Byte-Gate	149
11.4	Or - Port (2 in 1 Out)	149
11.5	And - Port (2 in 1 Out)	150
11.6	Or - Port (5 in 2 Out)	150
11.7	And - Port (5 in 2 Out)	151
11.8	Telegrammumwandler (0/1 Bit in 0-255 Byte)	151
11.9	Wert vergleichen	152
11.10	Speicherbildschirm 1 (RGB Wert)	152
11.11	Aufrufbildschirm 1 (RGB Wert)	153
11.12	RGB-Objekt	153
11.12.1	Arbeiten mit dem RGB-Objekt	153
11.12.2	Skript zur Steuerung der RGB LED-Farbe	155
11.12.3	Skript zum Auslösestatus vom RGB LED	155
11.13	Hysterese	156
11.14	Zufälliger Byte-Wert	156
11.15	Zyklischer Repeater (Verzögerung 60 Sekunden)	157
11.16	Stepper / Zähler Positive Eingabe	157
11.17	Stepper / Zähler Negative Eingabe	157
11.18	Stepper / Zähler zurücksetzen	158
11.19	Anschaltverzögerung (Schaltfläche auf „nur intern aktualisieren“ setzen)	158
11.20	Durchschnitt	158
11.21	Abschaltverzögerung	158
11.22	Timer Treppenaufgang (mit variablem Zeitobjekt)	158
11.23	Wertspeicher (in Speicher schreiben)	159
11.24	Wertspeicher (von Speicher holen)	159
11.25	Multiplexer (1 in / 3 out)	159

11.26	Rundungsfunktion mit Verwendung allgemeiner Funktionen	159
11.27	Datum und Zeit in KNX-Gruppenadressen schreiben.....	160
11.28	Daten auf Gruppen mit Tags schreiben.....	160
12	Anwendungshinweise	162
12.1.1	Liste mit aktuell herausgegebenen Anwendungshinweisen:.....	162
12.1.2	Verfügbarkeit der Anwendungshinweise	162

1 Produktsicherheit

1.1 Konnektivität

- IP-Verbindung
- USB
- RS232
- RS485
- WIFI über IP-Verbindung und WLAN-Router

1.2 Sicherheitsfunktionen

- Sicherheitsprotokoll <HTTPS://IP:Port>
- Netzwerksicherheit muss auf der entsprechenden Ebene festgelegt werden. spaceLYnk muss Teil eines sicheren Netzwerks mit eingeschränktem Zugriff sein. Im Falle einer Verbindung über das Internet wird die strikte Verwendung eines VPN- oder HTTPS-Kanals empfohlen.
- Sicherheitsmethode wird durch die Leistungsfähigkeit der anderen Netzwerkelemente (Firewall, Schutz vor Viren und Bedrohungen durch Schadsoftware) bestimmt.
- Eine regelmäßige Kennwort-Änderung, z.B. alle 90 Tage, wird dringend empfohlen. Das neue Kennwort muss sich vom alten unterscheiden und nicht ähnlich sein.

1.3 Bedienung

- Es wird dringend empfohlen, die Dateien mit den Sicherungskopien an einem sicheren Ort ohne Zugriff von unberechtigten Personen zu speichern.
- Die Eingabe des Kennworts sollte nicht zu einfach sein. Das Kennwort sollte Klein- und Großbuchstaben und Nummern enthalten, und mindestens 8 Zeichen lang sein.
- Für den Fall, dass Sie auf Cyber-Sicherheitsvorfälle oder Sicherheitslücken stoßen, melden Sie diese bitte auf folgender Website.
melden:
<http://www2.schneider-electric.com/sites/corporate/en/support/cybersecurity/contact-form.page>

1.4 Wartung

- Im Falle von Problemen oder Fragen in Bezug auf die Bedienung von spaceLYnk kontaktieren Sie bitte Ihren Lieferanten den Schneider Electric Helpdesk in Ihrem Land.
- Seien Sie sich bitte beim Remotezugriff auf Ihre lokales Netzwerk über das erhöhte Sicherheitsrisiko im Klaren.

1.5 Patchverwaltung

- Schritte zum Upgrade der Firmware finden Sie in Kapitel [3.3.1](#).
- Jedes Upgrade muss manuell durchgeführt werden. Vor einem Upgrade bitte eine Sicherheitskopie erstellen. Siehe Kapitel [3.2.7](#) zur Erstellung einer Sicherheitskopie.

1.6 Auf Werkseinstellungen zurücksetzen

- Eine Beschreibung zum Zurücksetzen auf Werkseinstellungen finden Sie in Kapitel [3.2.4](#).

1.7 Einschränkungen

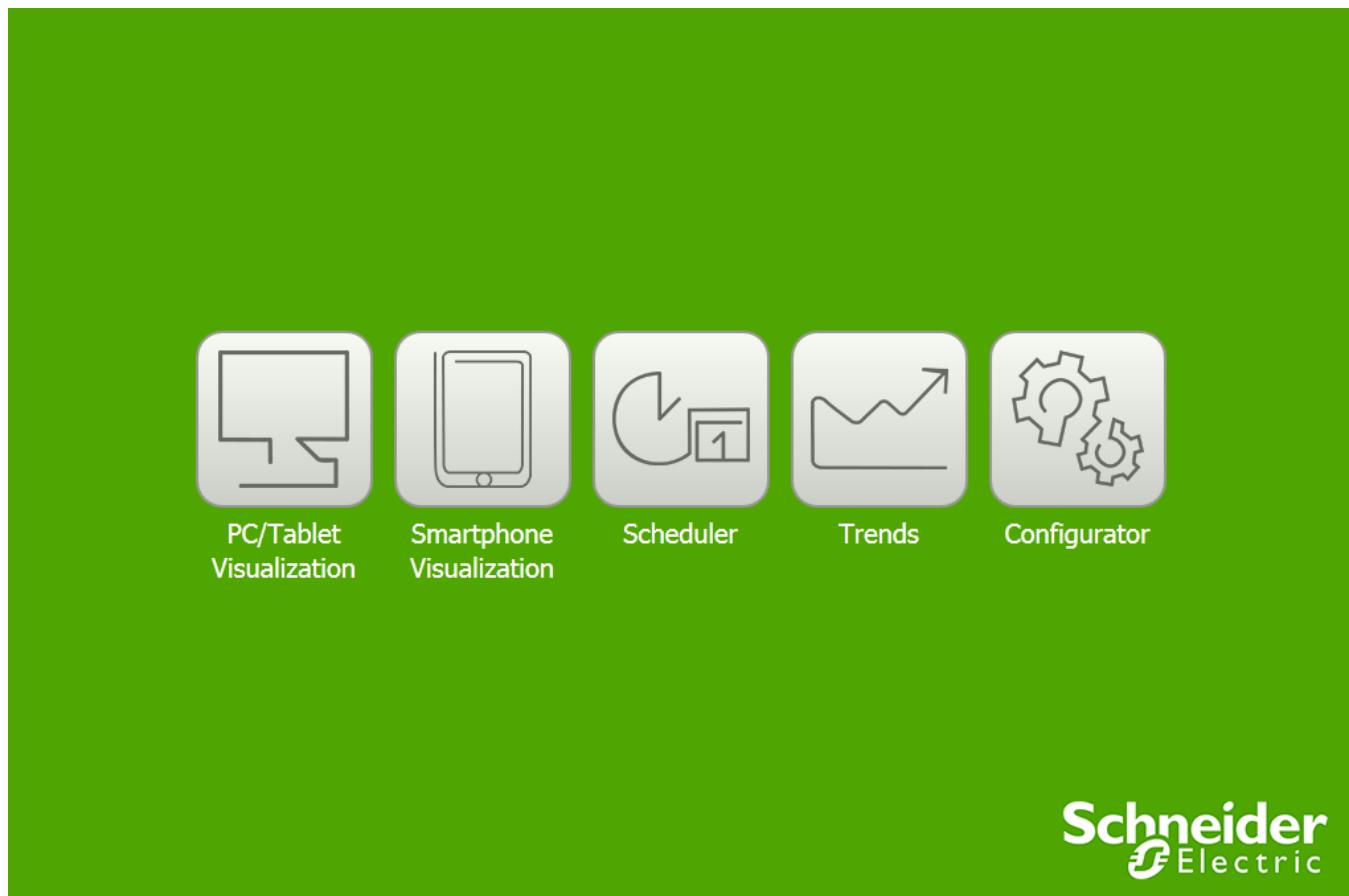
- Schneider Electric übernimmt keine Verantwortung für Schäden, die durch unqualifizierte Manipulation der spaceLYnk-Einstellungen entstehen und ist für keine Sicherheitsstörungen (z.B. durch unberechtigten Zugriff) verantwortlich.

2 Kurzanleitung

Die aufgelisteten Schritte helfen Ihnen, mit spaceLYnk vertraut zu werden.

1. Montieren Sie das Gerät auf einer DIN-Schiene.
2. Verbinden Sie die Buskabel (KNX, ModBus, und/oder RS232) und/oder das Flashlaufwerk
3. Verbinden Sie das 24V-Netzteil mit dem Gerät (positiver Leiter an die rote Klemme, negativer Leiter an die blaue Klemme).
4. Empfohlenes Zubehör - Stromversorgung REG/24V DC/0,4A, Artikel-Nr.:**MEG693003**.
5. Verbinden Sie das Ethernet-Kabel vom PC.
6. Standard IP-Adresse des spaceLYnk-Geräts lautet **192.168.0.10**. Ändern Sie die IP-Adresse des Computers auf den gleichen Bereich, z.B. **192.168.0.9**; Maske **255.255.255.0**.
7. Führen Sie Google Chrome oder Mozilla Firefox (für Windows), Safari (für OS X) aus, und navigieren Sie zu **192.168.0.10**.

HINWEIS: Internet Explorer wird nicht unterstützt.



2.1 Startseite

Startseite bietet eine Ansicht ähnlich dem Dashboard und zeigt auf die Schlüsselbereiche von spaceLYnk. Folgende Optionen finden Sie auf der Startseite.



PC-/Tablet-Visualisierung – Dieses Symbol navigiert zu einer Rich-Visualisierung mit Karten, in denen die einzelnen Objekte aufgeführt sind. Dies ist ideal für PCs, iPads und Android-Tablets (möglichst mit einem Bildschirm von 10", oder größer).



Smartphone-Visualisierung – Diese Symbol navigiert Sie zu einer vereinfachten Listen-Visualisierung, die für iPhone/iPod/iPad/Android-Smartphones/Android-Tablets (7" Bildschirm oder kleiner) entwickelt wurde. Alle im spaceLYnk- **Konfigurator** hinzugefügten Objekte sind in dieser Smartphone-Visualisierung standardmäßig sichtbar (wenn Verbergen in den **Smartphone-Optionen** nicht aktiviert ist).



Terminplaner – Dieses Symbol navigiert zu einer benutzerfreundlichen Oberfläche für den Endbenutzer zur Verwaltung von Planer-Aufgaben, wie beispielsweise die Spezifikation von Thermostat-Werten, die von Wochentag, Zeit und Ferien abhängig sind.



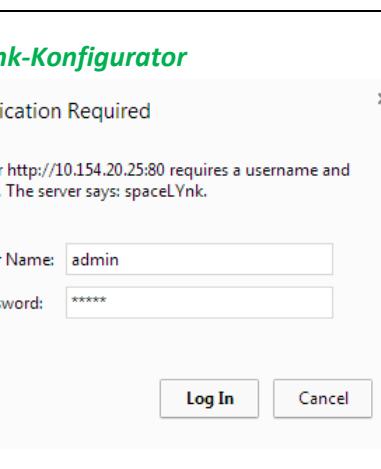
Trends – Dieses Symbol navigiert zu einer benutzerfreundlichen Anzeige der Trend-Protokolle, mit der Möglichkeit, Daten zwischen zwei unterschiedlichen Zeitpunkten zu vergleichen. Hier können Trends von bis zu 10 Jahren angezeigt werden.



Konfigurator – Dieses Symbol navigiert zur Benutzeroberfläche für Programmierung, Einstellungen und Konfiguration. Nur Administratoren haben auf diesen Bereich Zugriff.

HINWEIS: Der **Konfigurator** wird für Mobilgeräte nicht unterstützt, er kann nur im Desktop-Browser verwendet werden.

2.2 Standardkonfiguration

spaceLYnk-Konfigurator	Anmeldung	Kennwort
	admin	admin
HINWEIS: Es wird dringend empfohlen, das Standardkennwort zu ändern. Bei jedem Start des spaceLYnk- <i>Konfigurators</i> erscheint eine Erinnerung mit dem Standardkennwort.		
IP-Adresse im LAN	192.168.0.10	
Netzwerkmaske im LAN	255.255.255.0	

Ändern der IP-Einstellungen

Im Fenster **Dienstprogramme → System- → Netzwerkschnittstellen → des → Konfigurators** klicken Sie zur Änderung der IP-Einstellungen auf die spezifische Benutzeroberfläche.

Protokoll – Spezifisches, zur Adressierung verwendetes Protokoll:

- **Statische IP** – Statische IP-Adresse (Standard **192.168.0.10**).
- **DHCP** – DHCP-Protokoll zum Abrufen der IP-Konfiguration.

IP-Adresse – Die vom DHCP-Server erhaltene IP-Adresse. Dieses Feld erscheint nur, wenn eine IP-Adresse vergeben wurde. Es bleibt ansonsten verborgen.

Netzwerkmaske – Netzwerkmaske (Standard **255.255.255.0** (/24)).

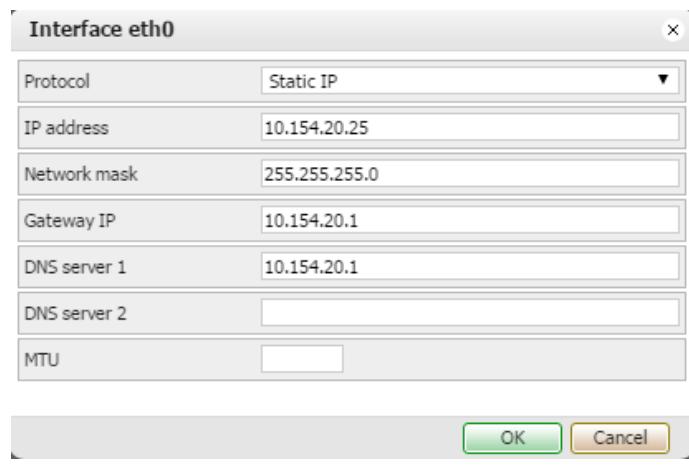
Gateway IP – Gateway-IP-Adresse.

DNS server 1 – Primäre IP-Adresse des DNS-Servers.

DNS server 2 – Sekundäre IP-Adresse des DNS-Servers.

Mtu – Maximale Übertragungseinheit. Die maximale Größe des Pakets, das im Kommunikationsprotokoll (standardmäßig 150) übergeben wird.

Bei Änderungen erscheint das **Apply changes** Symbol an der oberen rechten Ecke. Dies sollte angewendet werden, damit die Änderungen wirksam werden. spaceLYnk startet nach der Anwendung der Änderungen automatisch neu.



2.3 Erkennen der spaceLYnk IP-Adresse

Windows PC

Option 1:

Verwenden Sie das Dienstprogramm Service Browser, welches hier heruntergeladen werden kann:

<http://marknelson.us/attachments/2011/bonjour-windows/ServiceBrowserExe.zip>

Apple bonjour wird benötigt (Teil von iTunes):

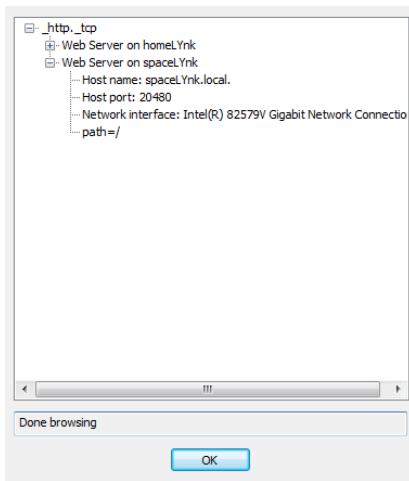
<http://support.apple.com/kb/DL999>

Option 2:

Tippen Sie folgendes in den Browser ein, wenn der Host nicht geändert, und nur **ein** spaceLYnk im Netzwerk vorhanden ist:

<http://spaceLYnk.local> - Firefox

<http://spaceLYnk.local> - Chrome



Linux PC

Das Dienstprogramm **Avahi** kann hier heruntergeladen werden:

www.avahi.org



Android



Die Gratis-App **ZeroConf Browser** kann bei Google Play heruntergeladen werden.



iOS/Mac OS



Der Gratis **Discovery bonjour browser** kann vom App Store heruntergeladen werden.

Installieren Sie für das iPad die iPhone/iPod-Version des Dienstprogramms.



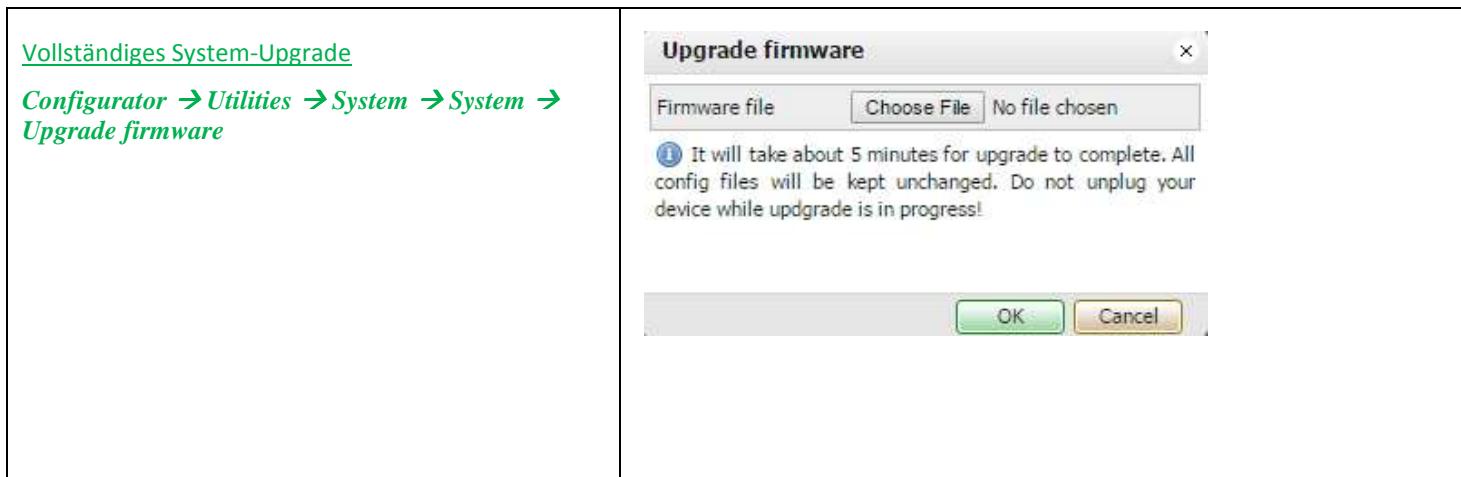
2.4 Firmware-Upgrade

HINWEIS: Sichern Sie bitte vor jedem Upgrade die Visualisierungen, Skripte und Objekte in **Configurator → Utilities → Backup**. Das Gerät reagiert während des Upgrades nicht, da ein Neustart durchgeführt wird.

Es wird nach jedem Upgrade empfohlen, den Browser-Cache zu leeren.

Verwenden Sie zum Software-Upgrade von spaceLYnk den Webbrowser. Die Firmware ist in Form von Bildern verfügbar und kann von der Support-Seite von SE office /**Shopping kiosk** heruntergeladen werden. (Verfügbar nur für SE-berechtigte Personen).

Schneider empfiehlt, Ihr Projekt nach jeder Projekt-Modifikation auf einem externen Laufwerk zu speichern.

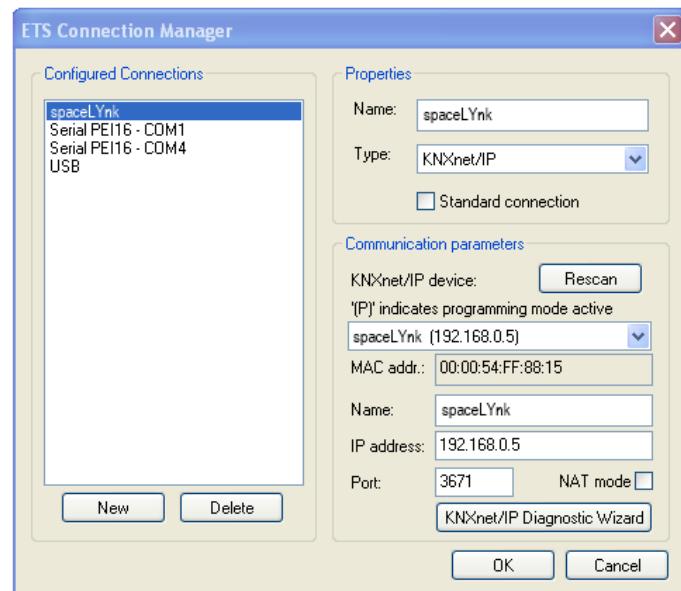


2.5 spaceLYnk für KNX/EIB Netzwerk-Konfigurationsverwaltung mit ETS3

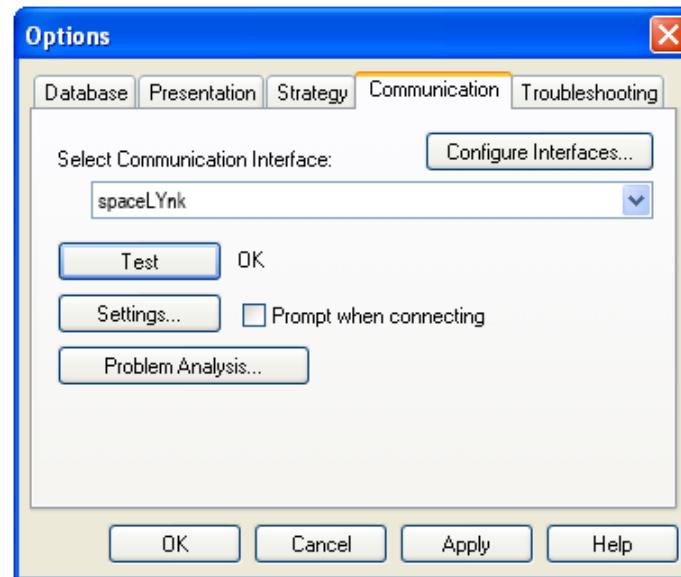
Zur Verwendung von spaceLYnk mit der KNXnet/IP-Funktionalität und zur Programmierung mit anderen KNX Bus-Geräten sollte das Gerät im **ETS-Verbindungs-Manager** hinzugefügt werden.

**Extras → Optionen → Kommunikation
→ Schnittstellenkonfiguration**

1. Geben Sie einen Namen für die Verbindung ein.
2. Wählen **Sie die Type, und wählen Sie KNXnet/IP aus dem Dropdown.**
3. Drücken Sie auf **Rescan** und wählen Sie spaceLYnk aus dem Dropdown.
4. Drücken Sie auf **OK**.

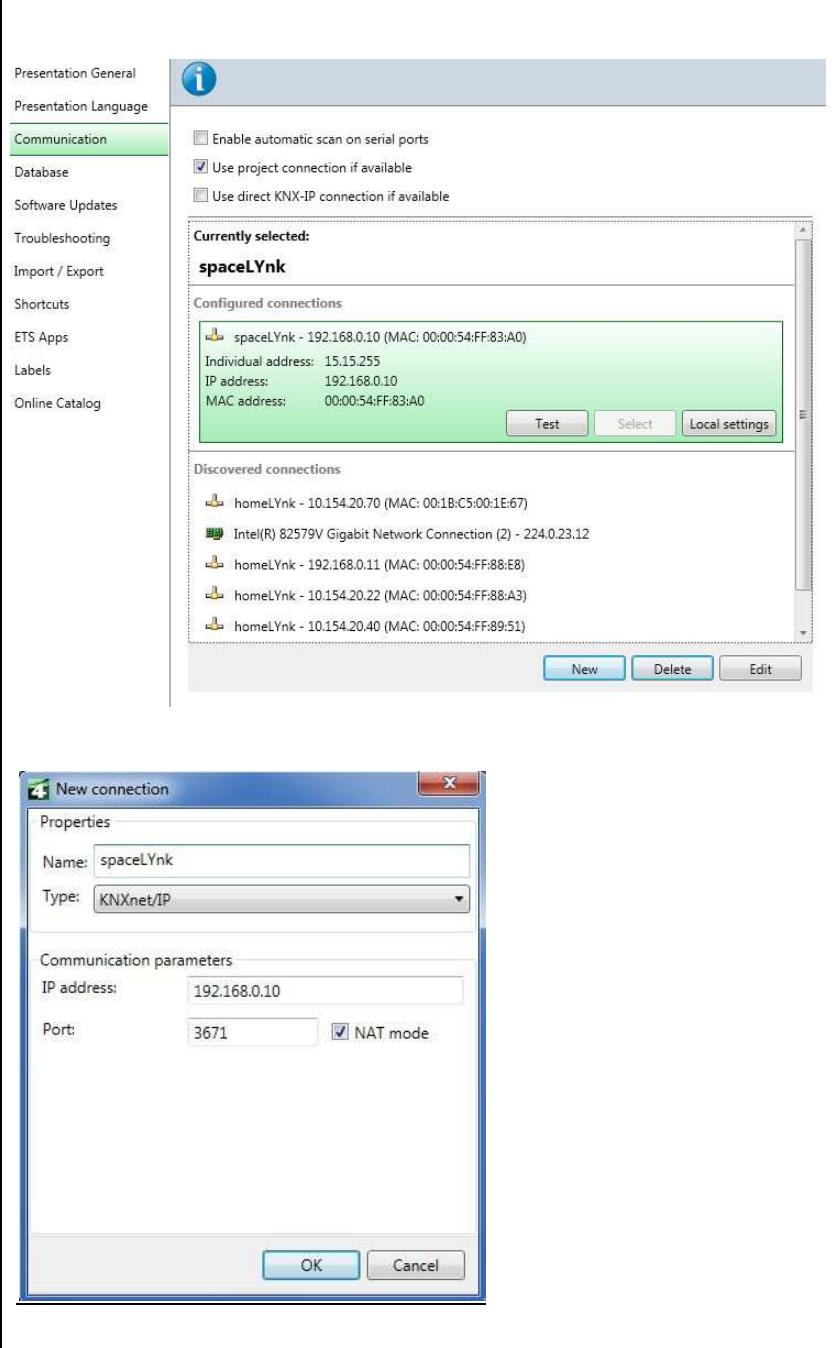


5. Wählen Sie die neu erstellte **Options → Communication** aus dem Dropdown im **Communication Interface**.
6. Zum Test der Kommunikation mit ETS drücken Sie auf **Test**.
7. Vergewissern Sie sich, dass der Bus-Status Online ist – drücken Sie die Schaltfläche in der ETS.



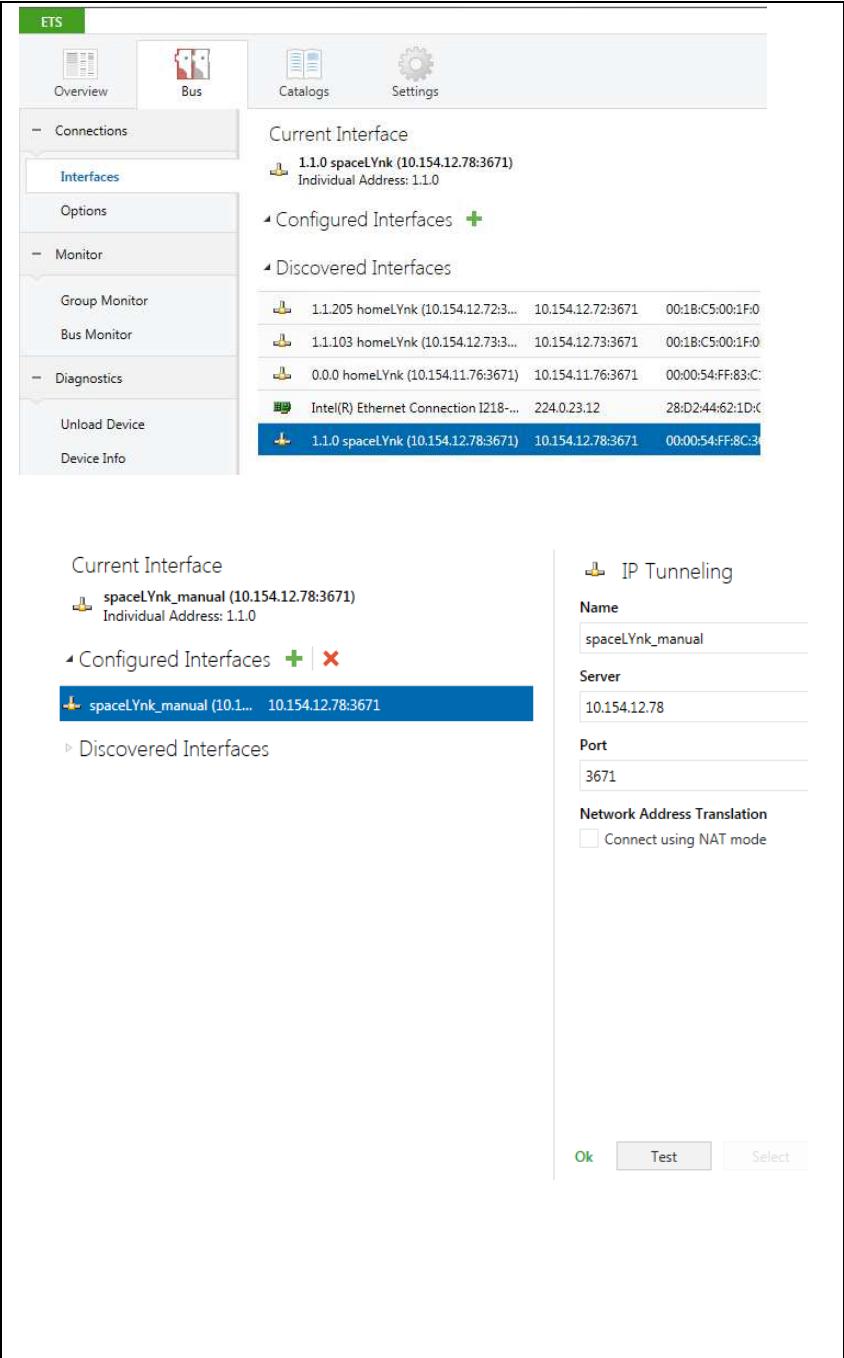
2.6 spaceLYnk für KNX/EIB Netzwerk-Konfigurationsverwaltung mit ETS4

Zur Verwendung von spaceLYnk mit der KNXnet/IP-Funktionalität und zur Programmierung mit anderen KNX Bus-Geräten sollte das Gerät im **ETS-Verbindungs-Manager** hinzugefügt werden. Die ETS-Programmierung durch spaceLYnk ist nur verfügbar, wenn die KNX IP-Funktionen aktiviert sind. Die Funktion „Bus-Überwachung“ ist in spaceLYnk nicht enthalten.

<p>Einstellungen→ Kommunikation</p> <p>Ein neu hinzugefügtes spaceLYnk wird automatisch ermittelt, wenn dieses mit dem gleichen Netzwerk verbunden ist wie der PC, auf dem die ETS4-Software ausgeführt wird.</p> <ol style="list-style-type: none">Wählen Sie Auswahl, um es zu den Konfiguriert-Verbindungen zu verschieben.Durch Drücken auf Lokale Einstellungen kann die spaceLYnk KNX individuelle Adresse und Maske festgelegt werden.Markieren Sie das Kontrollkästchen Projektverbindung verwenden, wenn verfügbar, um diese als Standard-Projektverbindung festzulegen.Wählen Sie die Option KNX-IP-Verbindung verwenden, wenn verfügbar, um eine direkte Kommunikation im IP-Netzwerk aufzubauen.Zum manuellen Hinzufügen von spaceLYnk drücken Sie auf Neu.Geben Sie einen Namen für das Gerät ein.Setzen Sie IP-Adresse, Port, und NAT-Modus (falls benötigt).Drücken Sie auf OK, um die Änderungen zu speichern.Zum Test der Kommunikation mit ETS drücken Sie auf Test.Drücken Sie auf Änderungen anwenden, damit diese wirksam werden.	
--	---

2.7 spaceLYnk für KNX/EIB Netzwerk-Konfigurationsverwaltung mit ETS5

Zur Verwendung von spaceLYnk mit der KNXnet/IP-Funktionalität und zur Programmierung mit anderen KNX Bus-Geräten sollte das Gerät im **ETS-Verbindungs-Manager** hinzugefügt werden. Die ETS-Programmierung durch spaceLYnk ist nur verfügbar, wenn die KNX IP-Funktionen aktiviert sind. Die Funktion „Bus-Überwachung“ ist in spaceLYnk nicht enthalten.

<p><u>Bus → Verbindungen → Schnittstellen</u></p> <p>Die spaceLYnk-Schnittstelle kann automatisch erkannt werden, wenn sich Ihr spaceLYnk auf dem selben Computer befindet, auf dem auch ETS5 ausgeführt wird. Sobald Ihr spaceLYnk erkannt wurde wählen Sie die Schnittstelle aus. Doppelklicken Sie dazu auf das Element in der Liste erkannte Schnittstellen.</p> <p>Folgen Sie den nachfolgenden Schritte, sollte Ihre Schnittstelle nicht erkannt worden sein:</p> <ol style="list-style-type: none">1. Klicken Sie auf das grüne + Symbol neben Konfigurierte Schnittstellen.2. Wählen Sie IP-Tunneling.3. Klicken Sie auf Neue Verbindung (0.0.0.0:3671), die in Konfigurierte Schnittstellen erstellt wird.4. Legen Sie in der Registerkarte Einstellungen auf der rechten Seite den Namen Ihrer Verbindung, Server (IP-Adresse von spaceLYnk) und Port fest.5. Wählen Sie die Schnittstelle, die Sie im vorherigen Schritt konfiguriert haben.6. Zum Test der Kommunikation mit ETS drücken Sie auf Test in der unteren rechten Ecke.7. Wenn der Test OK ist, wählen Sie die Schnittstelle als aktiv, indem Sie in der Liste Konfigurierte Schnittstellen auf das Element klicken.8. Die aktuelle Schnittstelle ist festgelegt.	 <p>ETS</p> <p>Overview Bus Catalogs Settings</p> <p>Connections</p> <p>Interfaces Options</p> <p>Monitor</p> <p>Group Monitor Bus Monitor</p> <p>Diagnostics</p> <p>Unload Device Device Info</p> <p>Current Interface</p> <p>1.1.0 spaceLYnk (10.154.12.78:3671) Individual Address: 1.1.0</p> <p>Configured Interfaces +</p> <p>Discovered Interfaces</p> <p>11.205 homeLYnk (10.154.12.72:3... 10.154.12.72:3671 00:1B:C5:00:1F:0 11.103 homeLYnk (10.154.12.73:3... 10.154.12.73:3671 00:1B:C5:00:1F:0 0.0.0 homeLYnk (10.154.11.76:3671) 10.154.11.76:3671 00:00:54:FF:83:C Intel(R) Ethernet Connection I218-... 224.0.23.12 28:D2:44:62:1D:C 1.1.0 spaceLYnk (10.154.12.78:3671) 10.154.12.78:3671 00:00:54:FF:8C:3</p> <p>Current Interface</p> <p>spaceLYnk_manual (10.154.12.78:3671) Individual Address: 1.1.0</p> <p>Configured Interfaces + x</p> <p>spaceLYnk_manual (10.1... 10.154.12.78:3671</p> <p>Discovered Interfaces</p> <p>IP Tunneling</p> <p>Name: spaceLYnk_manual</p> <p>Server: 10.154.12.78</p> <p>Port: 3671</p> <p>Network Address Translation: Connect using NAT mode</p> <p>Ok Test Select</p>
--	---

2.8 KNX und IP-Router-Einstellungen

Die spezifische KNX-Konfiguration befindet sich in:

**Konfigurator → Dienstprogramme → System → netzwerk
→ KNX-Verbindung.**

Registerkarte Allgemein

Modus: KNX-Verbindungsmodus.

- **TP-UART**- (standardmäßig integriert) –
Physikalische Verbindung über Twisted-Pair.
- **EIBnet/IP Tunneling**- Zwischen EIBnet/IP-Server und dem Client wird eine Punkt-zu-Punkt-Verbindung über einen Kommunikationskanal aufgebaut.
- **EIBnet/IP Tunneling (NAT Modus)** –
Tunnelmodus-Modus mit Netzwerkadressübersetzung.
- **EIBnet/IP Routing**- Austausch von KNX/EIB-Telegramme zwischen verschiedenen EIBnet/IP-Servern. Ein EIBnet/IP-Server leitet nur die Telegramme weiter, die den Filterkriterien entsprechen. Das Routing verwendet Multicast-Telegramme.

ACK alle Gruppentelegramme: Bei Auswahl erkennt spaceLYnk jedes Gruppentelegramm an.

KNX-Adresse: KNX-physikalische Adresse des Geräts.

KNX IP-Funktionen: Verwenden Sie dieses Gerät mit KNX IP-Funktionen. Beispielsweise KNXnet/IP-Netzwerkkonfiguration.

Multicast-IP: Multicast-IP-Adresse:

Multicast-TTL: Multicast-TTL (Gültigkeitsdauer).

Maximale Telegramme in Warteschlange: Anzahl der maximalen Telegramme in der Warteschlange.

HINWEIS: Wenn der KNX TP nicht mit einem Gerät verbunden ist, sollte der **Routing**-Modus zur korrekten Aktualisierung der Gruppenadressen verwendet werden.

Nach den Änderungen erscheint das Symbol an der oberen rechten Ecke.

Dies sollte angewendet werden, damit die Änderungen wirksam werden. spaceLYnk startet nach der Anwendung der Änderungen automatisch neu.



2.9 Visualisierung erstellen für iPad/PC

2.9.1 Objekte importieren

Zur Fertigstellung des ETS3-Projekts erstellen Sie eine *.ESF-Datei von ETS durch:

Datei → extrahieren data → Export zu OPC-Server

Importieren *.ESF-Datei in

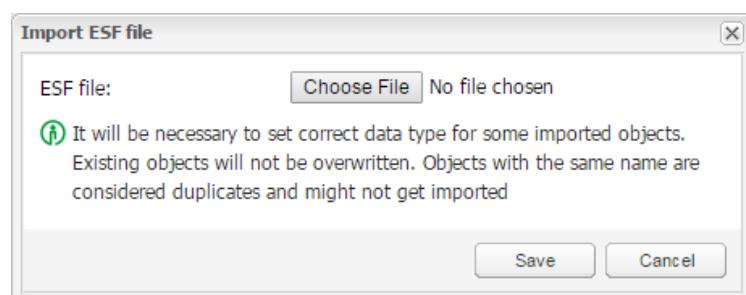
Konfigurator → Dienstprogramme → ESF-Datei importieren

Zur Fertigstellung des ETS4 oder EST5-Projekts, erstellen Sie eine *.ESF-Datei von ETS-Projekt durch:

Extras → Export OPC

Importieren *.ESF-Datei in:

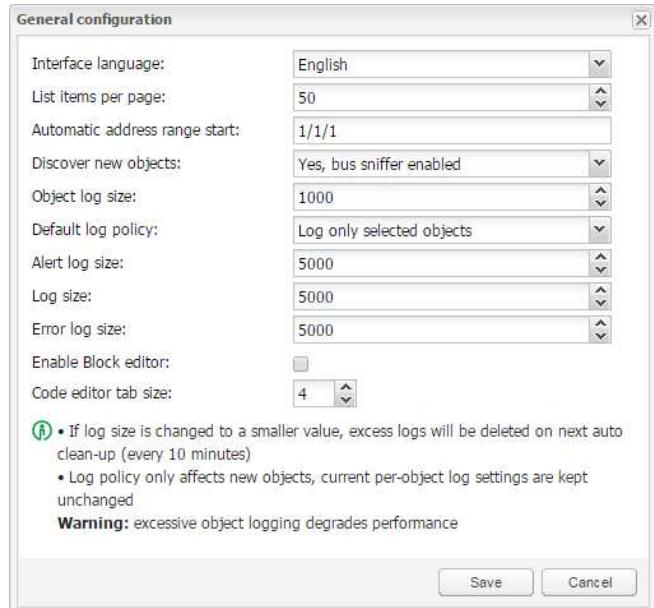
Konfigurator → Dienstprogramme → ESF-Datei importieren



Oder verbinden Sie spaceLYnk mit dem KNX-Bus. Dadurch werden **Objekte** nach der Aktivierung (KNX-Telegramm versendet) automatisch in der Registerkarte Objekte erkannt.

Diese Option kann in Konfigurator Dienstprogramme **Allgemeine Einstellung → Neue → Objekte erkennen → aktiviert oder deaktiviert werden.**

HINWEIS: Objekte können außerdem manuell hinzugefügt werden.



2.9.2 Erstellen Sie eine Gebäude-/Etagen-Struktur und fügen Sie der Karte Objekte hinzu

Navigieren Sie zu **Konfigurator → Vis.structure**

Vis. In Struktur können alle Gebäudeebenen und Visualisierungspläne erstellt werden. Zusätzlich können dort Layouts und Widgets für Visualisierungspläne erstellt werden.

Beim Start eines neuen Projekts sind nur Ordner für **Layouts** und **Widgets** sichtbar. Das Hinzufügen neuer Ebenen ermöglicht die Definition eines spezifischen **Plans** des Apartments. **Layouts** und **Widgets** sind zusätzliche Tools, die für die einfache Visualisierung nicht zwingend sind. Sie können in vielen anderen **Plänen** definiert und eingebunden werden.

Ebenen

Zum Hinzufügen einer neuen Ebene drücken Sie auf die Schaltfläche. Die Hauptebene ist normalerweise der Projektname (zusätzliche Ebenen können später hinzugefügt werden).

Levels / Plans								
Name	Visible	Description	Duplicate	Move up	Move down	Add / Import	Export	
SE spaceLYnk...	✓		⊕	↑	↓	⊕	⊖	✖
Overview	PC/Tablet, Smar...		⊕	↑	↓	⊕	⊖	✖
Main Office	PC/Tablet, Smar...		⊕	↑	↓	⊕	⊖	✖
Meeting R...	PC/Tablet		⊕	↑	↓	⊕	⊖	✖
Facility Ma...	PC/Tablet		⊕	↑	↓	⊕	⊖	✖
Garage	PC/Tablet, Smar...		⊕	↑	↓	⊕	⊖	✖
Trend Gra...	PC/Tablet		⊕	↑	↓	⊕	⊖	✖
Schedulers	PC/Tablet		⊕	↑	↓	⊕	⊖	✖
Help	PC/Tablet		⊕	↑	↓	⊕	⊖	✖

Zweite Ebene

Drücken Sie auf neben der Hauptebene, um **+** zusätzliche Ebenen hinzuzufügen.



Wählen Sie **Zweite Ebene hinzufügen** und geben Sie Namen und Sortierreihenfolge an.

Durch Drücken auf das Duplizieren-Symbol neben der Ebene kann jede der Ebenen mit Unterebenen und Plänen dupliziert werden.

Pläne

Zum Hinzufügen eines Plans drücken Sie auf und wählen **Plan hinzufügen**.

Übergeordnet: Übergeordnetes Verzeichnis.

Name: Name des Plans.

Plangröße: Größe des Plans. Benutzerdefiniert oder von der Liste vordefiniert.

Layout: Layout für den Plan. Alle **Objekte** des Layouts werden, einschließlich Hintergrundfarbe und Planbild, wenn nicht separat definiert, in den Plan dupliziert.

PC/Tablet Visualisierung [Show, Show, make default, Hide]: Sichtbarkeit für diesen bestimmten Plan in der PC/Tablet-Visualisierung.

Smartphone Visualisierung [Show, Show, make default, Hide]: Sichtbarkeit für diesen bestimmten Plan in der Smartphone-Visualisierung.

Pin-Code: Festlegen des Codes, der zum Zugriff auf diesen bestimmten Plan der Visualisierung benötigt wird. Wenn nicht definiert, wird beim Zugriff auf den Visualisierungsplan kein Code abgefragt.

Primäres Hintergrundbild: Wählen Sie den Hintergrund, der zuvor Vis hinzugefügt wurde.

Grafiken → Bilder/Hintergründe.

Sekundäres Hintergrundbild: Wählen Sie den sekundären Hintergrund, der zuvor Vis hinzugefügt wurde. **Grafiken → Bilder/Hintergründe** für Parallaxen-Projektion.

Hintergrundfarbe: Wählen Sie die Hintergrundfarbe des Plans für die PC/Tablet-Visualisierung.

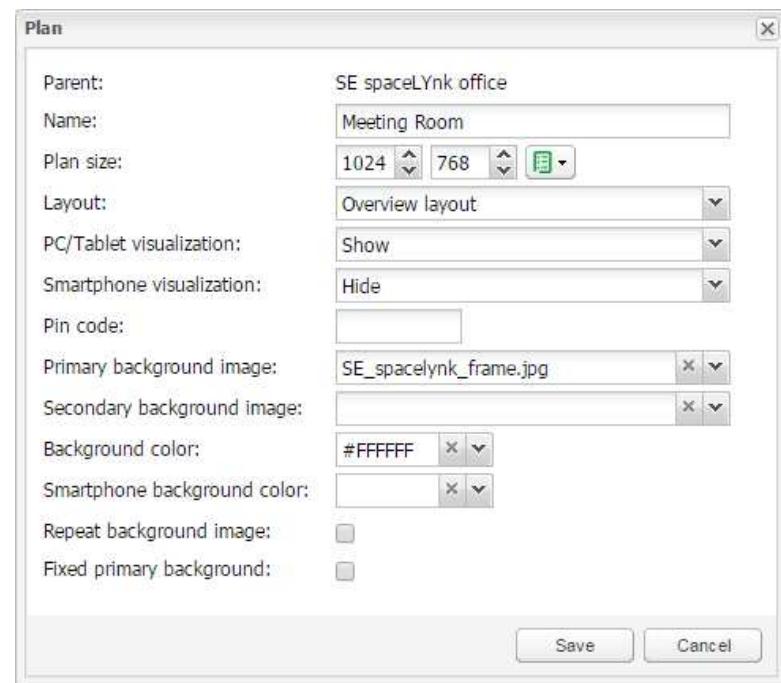
Smartphone-Hintergrundfarbe: Wählen Sie die Hintergrundfarbe des Plans für die Smartphone-Visualisierung.

Hintergrundbild wiederholen: Zur einmaligen Anzeige des Bildes, oder zur wiederholten Anzeige, um den kompletten Plan auszufüllen.

Fester, primärer Hintergrund: Primärer Plan in der Parallaxen-Projektion ist statisch.

HINWEIS: Durch Drücken auf das Duplizieren-Symbol neben dem Plan kann jeder Plan zusammen mit allen Komponenten in einem Plan dupliziert werden .

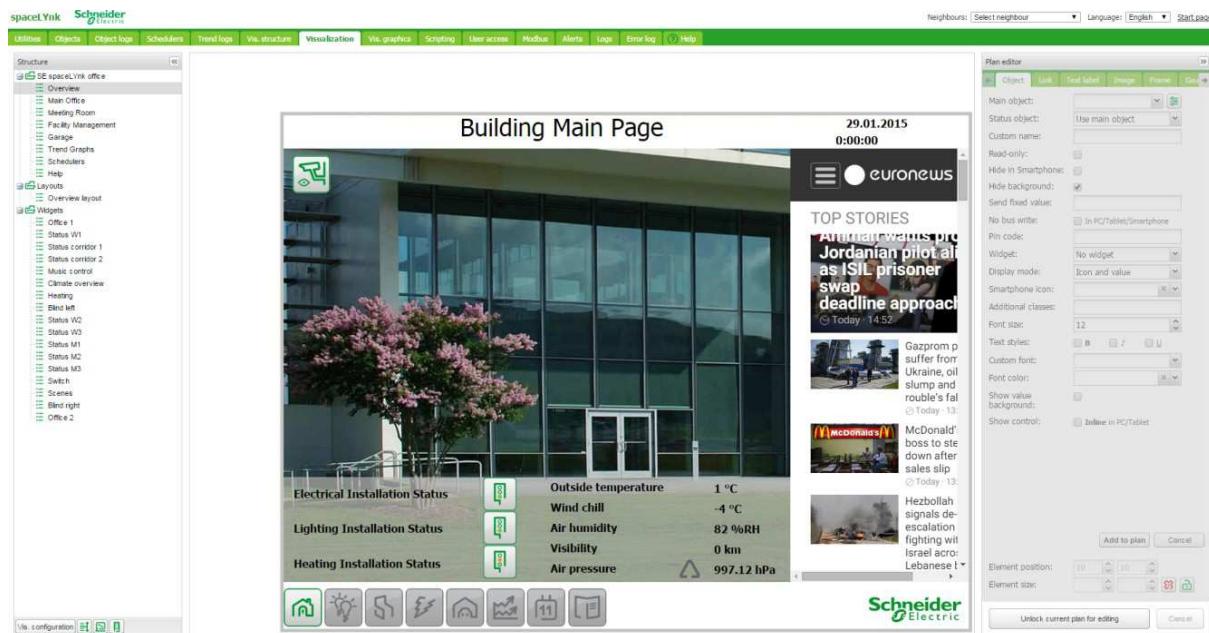
Levels / Plans		Layouts / Widgets						
Name	Visible	Description	Duplicate	Move up	Move down	Add / Import	Export	
SE spaceLYnk...								
Overview	PC/Tablet, Smar...							
Main Office	PC/Tablet, Smar...							
Meeting R...	PC/Tablet							
Facility Ma...	PC/Tablet							
Garage	PC/Tablet, Smar...							
Trend Gra...	PO/Tablet							
Schedulers	PC/Tablet							
Help	PC/Tablet							



2.9.3 Der Neu erstellten Visualisierungskarte Objekte hinzufügen

Navigieren Sie zu **Konfigurator → Visualisierung**.

Nachdem die **Ebene** und die Struktur des Plans in der Registerkarte **Vis.structure** definiert wurden, können diese in der Registerkarte **Visualisierung** sichtbar gemacht werden. In diesem Abschnitt können gesteuerte und überwachte Objekte hinzugefügt und verwaltet werden. Durch Drücken auf den linken oder rechten Pfeil können beide Randleisten minimiert werden. Dies verleiht der Karte, speziell bei kleineren Bildschirmen, eine bessere Sichtbarkeit.



Durch Klicken auf Aktuellen Plan zur Bearbeitung **entsperren können der Karte existierende Objekte hinzugefügt werden**. Nach der Definition der Objektparameter drücken **Sie auf Zu Plan hinzufügen**, und das neu erstellte Objekt erscheint. Dieses Objekt kann dann an die gewünschte Stelle verschoben werden. Im Bearbeitungsmodus funktioniert das Objekt jedoch nicht. Nach dem Hinzufügen aller nötigen Objekte drücken Sie auf **Speichern und laden** den Plan erneut, damit die Objekte visualisiert werden können.

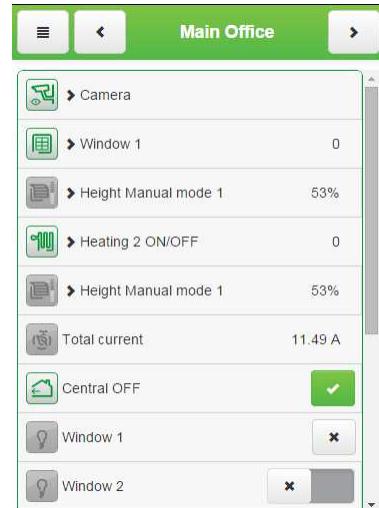
2.9.4 Starten der Visualisierung auf einem Smartphone (in diesem Fall iPod)

Befolgen Sie dazu die folgenden Schritte:

1. Vergewissern Sie sich, dass der iPod drahtlos mit dem spaceLYnk verbunden ist (durch einen separaten Zugriffspunkt – Drahtlosrouter).
2. Geben Sie die spaceLYnk-IP (Standard **192.168.0.10**) in den Webbrowser des iPod ein.



3. Klicken Sie auf das Symbol **Smartphone Visualisierung**.
4. Die Verknüpfung zur Anwendung kann gespeichert werden.
Legen Sie zum einfachen Zugriff einen Shortcut für den iPod an.



2.9.5 Starten der Visualisierung auf einem PC, Tablet oder jeglichem anderen Touch-Gerät mit großem Bildschirm

Befolgen Sie dazu die folgenden Schritte:

1. Vergewissern Sie sich, dass PC/Tablet auf spaceLYnk zugreifen können und geben Sie die IP in den Browser ein (Standard **192.168.0.10**).
2. Klicken Sie auf **PC/Tablet-Visualisierung**.
3. Wählen Sie den gewünschten **Plan** aus.
4. Zur besseren Ansicht kann die Randleiste durch Drücken auf das Symbol minimiert werden.



3 Erweiterte Anleitung

3.1 Hauptseite Konfigurator

Hauptseite Konfigurator - Leiste oben:



Nachbarn - Umschalten zum nächsten spaceLYnk im gleichen Netzwerk. Diese Auswahl erscheint nur, wenn andere homeLYnks oder spaceLYnks erkannt werden.

Sprache - Wechseln Sie die Sprache der operativen Benutzeroberfläche zu Englisch, Tschechisch, Dänisch, Niederländisch, Französisch, Deutsch, Italienisch, Portugiesisch, Russisch, Spanisch oder Türkisch.

Startseite - Verknüpfung zur Haupt-Visualisierungsseite.

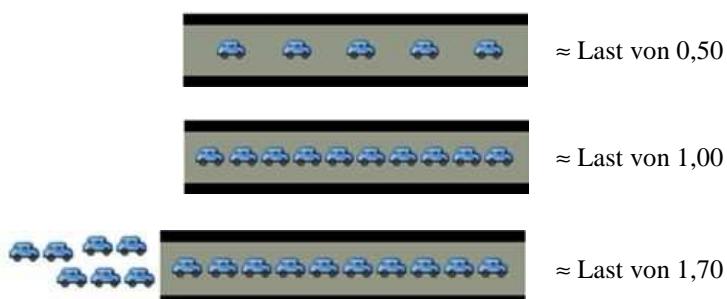
Hauptseite Konfigurator - Leiste unten:



Version: 1.1 - Dies ist die aktuelle Firmware-Version des spaceLYnk.

CPU/IO: 0.59 0.44 0.25, Speicher 11% - Laden der Durchschnittszahlen **0,59 0,44 0,25** - repräsentieren Mittelwerte über progressiv längere Zeiten (eine, fünf und fünfzehn Minuten-Mittelwert). Je geringer die Zahl, desto besser.

Datenverkehr-Analogie zu Prozessen überbrücken:

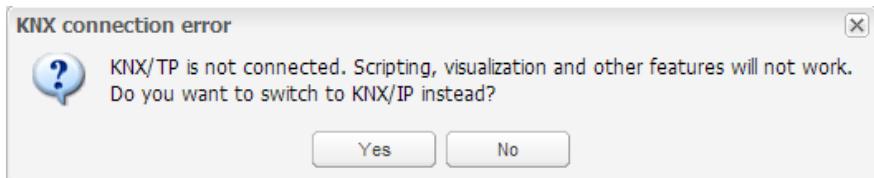


HINWEIS: Überprüfen Sie Ihre aktiven Tasks, falls die Last den Level **0,70** übersteigt!

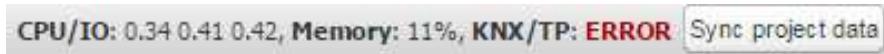
Speicher = $\frac{\text{Usedmemory-Buffered-Cached}}{\text{Totalsystemmemory}}$ (mindestens belegter Speicher in %)

Siehe **System / Status / Systemstatus / Speicherauslastung**. Auf Linux-Terminologie achten. Linux nennt zwischengespeicherte und gepufferte Speicher „verwendet“, selbst wenn dies für neue Anwendungen als „frei“ verstanden werden könnte.

KNX/IP: Bei jedem Öffnen des Konfigurators prüft spaceLYnk, ob der KNX-Bus verbunden ist. Bei Nichtverbindung erscheint folgende Fehlermeldung: Scripting, Visualisierung und andere Funktionen funktionieren nicht. Möchten Sie zu KNX/IP wechseln?



Die ausgewählte Verbindung und deren Status wird an der unteren rechten Ecke angezeigt:



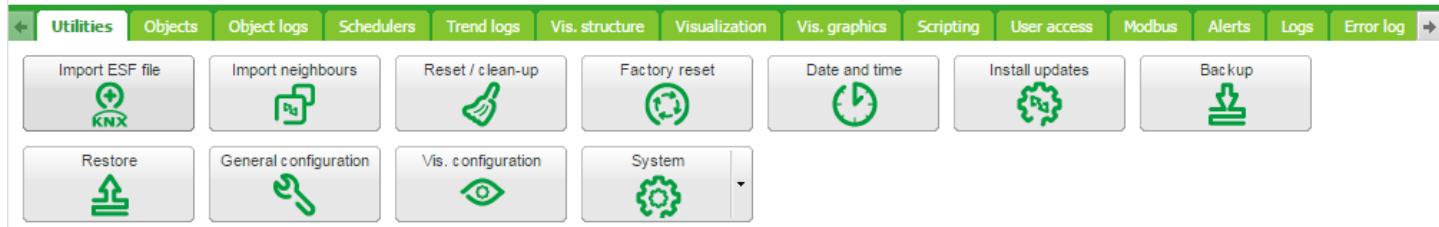
HINWEIS: Sobald der KNX-Bus unter **System → Network → KNX connection → Mode** verbunden ist, muss die KNX-Schnittstelle zurück zu TP-UART geändert werden. Die Änderung der KNX-Schnittstelle muss durch einen Neustart von **Apply changes** spaceLYnk bestätigt werden (manuell oder durch Drücken der Schaltfläche).

Projektdaten synchronisieren: Diese Schaltfläche ist bei größeren Änderungen im Projekt nützlich. Wenn gedrückt, wird das Projekt unverzüglich mit der microSD-Karte synchronisiert.

Die automatische Synchronisation wird nur alle 30 Minuten durchgeführt und nicht gespeicherte Daten gehen evtl. verloren.

3.2 Dienstprogramme

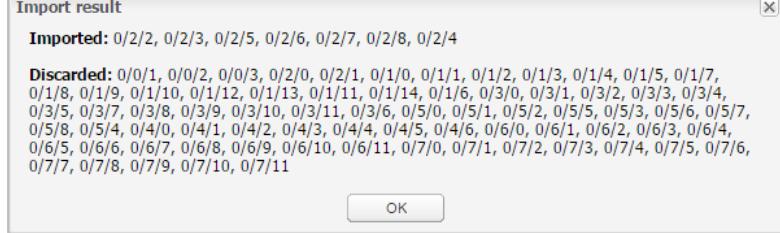
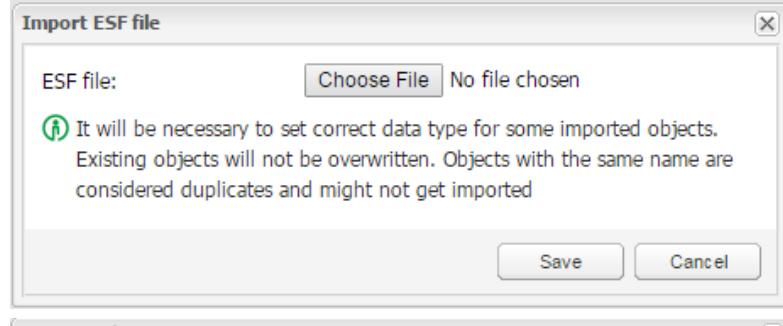
Nachfolgend ein Screenshot der Registerkarte Dienstprogramme.



3.2.1 Importieren *.ESF-Datei

Importiert die ETS-Objektdatei. Zum Import von Objekten ist es wichtig, die korrekten Datentypen zu setzen. Bestehende Objekte werden nicht überschrieben. Objekte mit dem gleichen Namen werden als Duplikate angesehen und evtl. nicht importiert. Nach jedem Import wird eine Liste der importierten und verworfenen Objekte angezeigt.

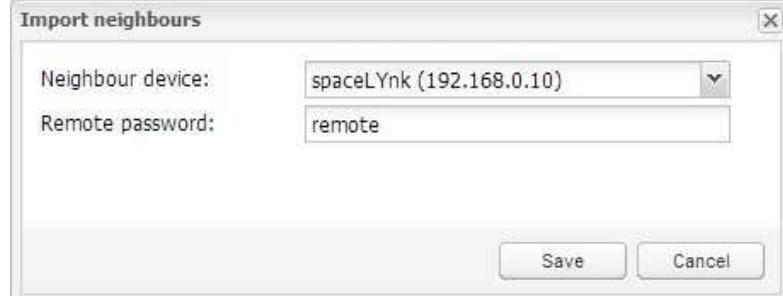
(Siehe Kapitel 1.8.1 für ETS3, ETS4 und ETS5 Importdetails.)



3.2.2 Nachbarn importieren

Bei mehr als einem spaceLYnk im gleichen IP-Netzwerk erscheint in **Dienstprogramme** die Schaltfläche **Nachbarn importieren**. Dies ermöglicht das Importieren der exportierten Objekte von einem anderen spaceLYnk. Das System fragt nach dem **Remotekennwort** des zweiten Geräts, von dem die Daten exportiert werden.

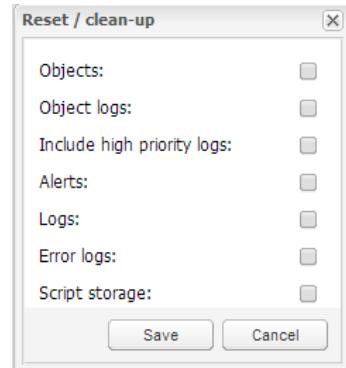
Für **Objekte**, die zwischen den spaceLYnks geteilt werden, muss die Export-Option aktiviert werden. Eine Aktivierung macht diese Objekte über BACNet und Remotedienste (XML/JSON) sichtbar.



3.2.3 Zurücksetzen / Aufräumen

Löscht ausgewählte Elemente vom spaceLYnk. Wenn Sie **Objekte** auswählen werden diese auch vom Visualisierungsteil gelöscht.

HINWEIS: Vor dem Fortfahren mit Zurücksetzen / Aufräumen alle wichtigen Daten sichern.



3.2.4 Auf Werkseinstellungen zurücksetzen

Löschen Sie alle Konfigurationen und setzen Sie auf Werkseinstellungen zurück. Die Systemeinstellungen bleiben erhalten.

Zur Durchführung einer **kompletten** Rücksetzung auf Werkseinstellungen drücken Sie zweimal lange auf die Schaltfläche ZURÜCKSETZEN, die sich auf der Rückseite des spaceLYnk-Geräts befindet (für 10 Sekunden drücken, loslassen, und erneut für 10 Sekunden drücken).

Die IP-Adresse nach einem Zurücksetzen auf HW-Werkseinstellungen mit der Schaltfläche HW ist immer 192.168.0.10.

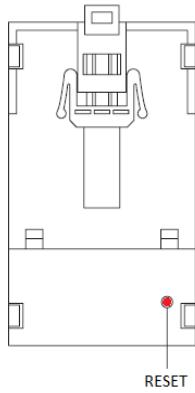
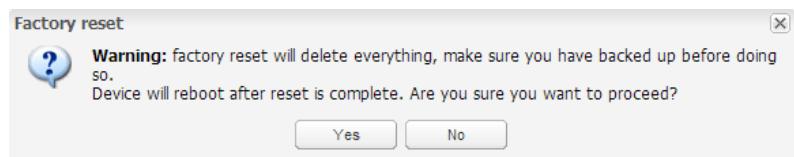
Bei einem Zurücksetzen auf Werkseinstellungen mit der Schaltfläche SW im Hauptmenü bleibt die

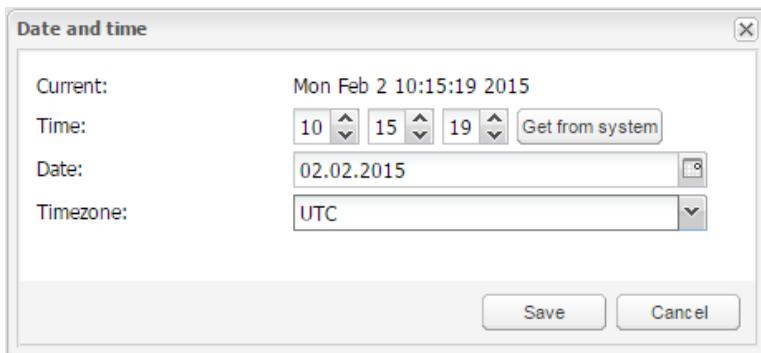
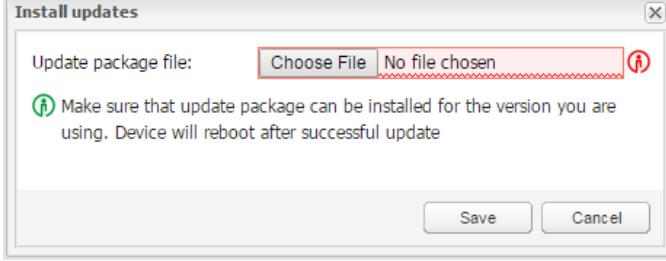
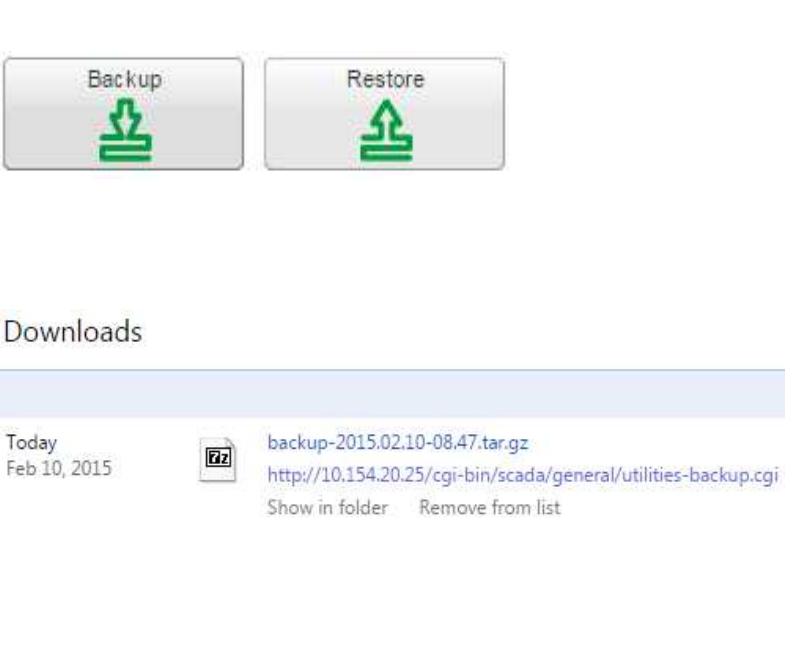


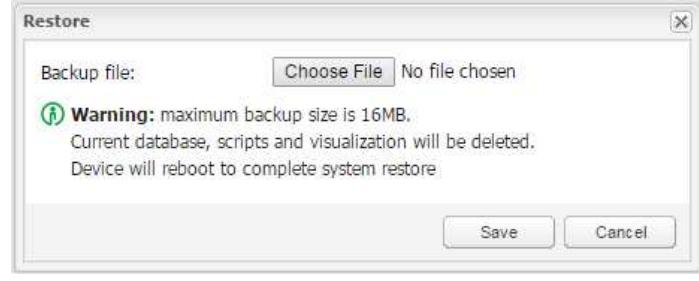
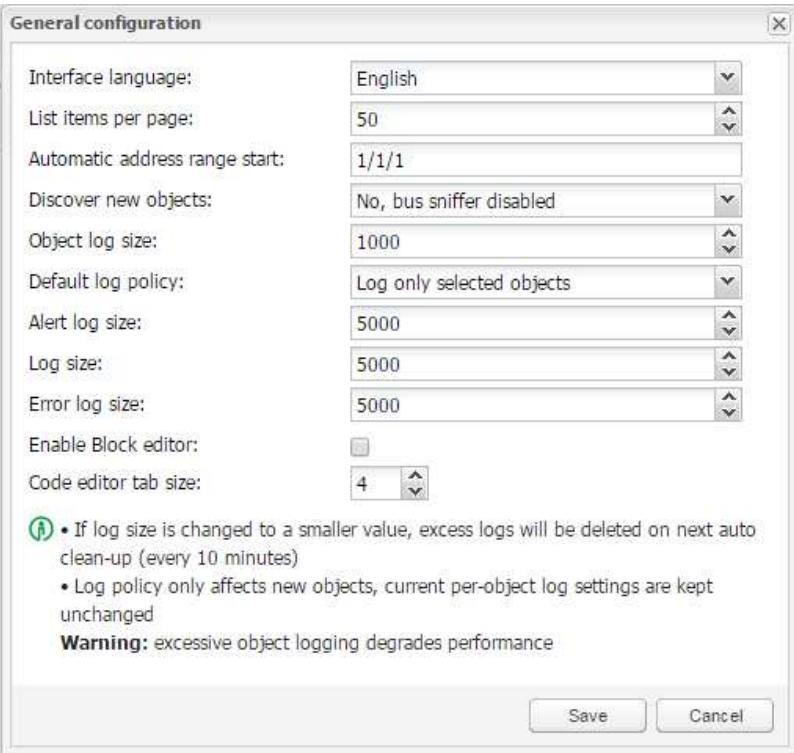
IP-Adresse unverändert und auch die Visualisierungsgrafik bleibt erhalten.

Für die Standard-IP-Adresse mit erhaltenem Projekt drücken Sie die HW-Zurücksetzen-Schaltfläche für 10 Sekunden.

HINWEIS: Ein komplettes Zurücksetzen mit der Hardware-Schaltfläche wird speziell für Situationen durchgeführt, in denen auf spaceLYnk aufgrund falscher Einstellungen nicht zugegriffen werden kann.



<h3>3.2.5 Datum und Zeit</h3> <p>Network Time Protocol (NTP) ist implementiert. Mit einer Internetverbindung aktualisiert spaceLYnk die Zeit automatisch von den in Dienstprogramme/System/Services/NTP-Client definierten Servern:</p> <ul style="list-style-type: none">0.europe.pool.ntp.org1.europe.pool.ntp.org2.europe.pool.ntp.org3.europe.pool.ntp.org <p>HINWEIS: Es ist wichtig, die aktuelle Zeitzone auszuwählen.</p> <p>Ohne Internetverbindung klicken Sie auf Vom System erhalten, um die Zeit vom PC zu übernehmen.</p> <p>HINWEIS: Ohne Stromzufuhr verwaltet spaceLYnk die Einstellungen für Zeit und Datum nur für einen begrenzten Zeitraum (ca. 5 Minuten).</p>	
<h3>3.2.6 Updates installieren</h3> <p>Für Teilupdates oder das Hinzufügen von vorgefertigten Lösungen.</p> <p>Die Installation der spaceLYnk-Update-Datei *.lmup. spaceLYnk startet das System nach einem erfolgreichen Update neu.</p>	
<h3>3.2.7 Sicherung</h3> <p>Sichern Sie Objekte, Trends, Protokolle, Skripte, Symbole, Hintergründe, Visualisierungen und KNX-Filtertabellen in der Datei backup-dd.mm.yyyy-hh.mm.tar.gz (bei der Erstellung der Sicherung werden Uhrzeit und Datum von spaceLYnk verwendet).</p> <p>Die Sicherungsdatei wird im Download-Ordner des Browsers abgelegt.</p> <p>Die Datei kann umbenannt werden, um der Projektstruktur zu entsprechen.</p> <p>HINWEIS: Maximale Größe der Sicherung ist 16 MB. Prüfen Sie nach der Erstellung die Größe der Sicherung. Sollte diese 16 MB übersteigen, können Sie diese nicht speichern. Systemkonfiguration, Netzwerkeinstellungen, Kennwörter oder KNX-Einstellungen werden nicht gesichert. Die Filtertabelle wird gesichert.</p>	 <p>Backup Restore</p> <p>Downloads</p> <p>Today Feb 10, 2015 backup-2015.02.10-08.47.tar.gz http://10.154.20.25/cgi-bin/scada/general/utilities-backup.cgi Show in folder Remove from list</p>

<h3>3.2.8 Wiederherstellen</h3> <p>Stellt die Konfiguration von der Sicherung wieder her.</p>	
<h3>3.2.9 Allgemeine Konfiguration</h3> <p>Menüsprache: Menüsprache (Englisch, Tschechisch, Dänisch, Niederländisch, Französisch, Deutsch, Italienisch, Portugiesisch, Russisch, Spanisch und Türkisch).</p> <p>Elemente pro Seite auflisten: Zählen der Zeilen pro Seite, zum Beispiel: Objekte, Objekt-Protokolle, Warnungen usw. (Maximalwert ist 50).</p> <p>Automatischer Adressbereichsanfang: Neu hinzugefügte Gruppenobjekte beginnen mit der Adressierung von einem definierten Bereich.</p> <p>Neue Objekte erkennen: KNX-Objekt-Sniffer ist aktiviert. Wenn JA ausgewählt ist, erscheinen alle neuen Objekte in der Objektliste.</p> <p>Größe Objektpfotokoll: Anzahl der Objektprotokolle. (Maximalwert ist 10000).</p> <p>Standard-Protokollrichtlinie: Hier kann der Protokollstatus für alle Objekte oder nur für geprüfte Objekte ausgewählt werden.</p> <p>Größe Warnprotokoll: Anzahl der protokollierten Warnungen. (Maximalwert ist 5000).</p> <p>Protokollgröße: Anzahl der Protokolle. (Maximalwert ist 5000).</p> <p>Größe Fehlerprotokoll: Anzahl der protokollierten Fehler. (Maximalwert ist 5000).</p> <p>Block-Editor aktivieren: Aktivieren/deaktivieren der Funktion Block-Editor in Scripting.</p> <p>Größe Code-Editor-Tab: Das Drücken auf TAB im Scripting-Editor resultiert im Einfügen einer definierten Anzahl an Leerzeichen</p>	 <p>HINWEIS:</p> <p>Bei der Änderung von Elemente pro Seite auflisten oder Sprachparameter muss spaceLYnk neu gestartet werden. (Löschen Sie den Browser-Cache).</p> <p>spaceLYnk behält die Protokollobjekte oberhalb des Limits für 15 Minuten. Nach Ablauf der Zeit werden alle Datensätze über dem Limit gelöscht. Dies ist wichtig, wenn große Mengen an Daten in Zeit Protokolliert werden.</p> <p>(Übermäßige Protokollierung von Objekten beeinträchtigt die Leistung von spaceLYnk.)</p>

3.2.10 Vis.configuration

PC/Tablet-Randleiste: [Gedockt anzeigen / Als Überlagerung anzeigen / Verbergen] Aktivierung der Randleiste in Visualisierung gedockt/mit Option auto-verbergen/ verborgen.

PC/Tablet-Ansicht: [Pläne links ausrichten, keine Größeneinschränkungen / Pläne zentrieren, Größeneinschränkung /Pläne zentrieren, automatische Größe aktivieren / horizontal zentrieren, automatische Anpassung Größe]

HINWEIS: Automatische Anpassung funktioniert nur in Webbrowsers mit Web Kit-Engine (Chrome, Safari) und Firefox.

PC/Tablet Seitenübergang: [Kein Übergang / X umkehren / Y umkehren / Verkleinern / Vergrößern / Nach oben ziehen / Nach unten ziehen / Nach links ziehen / Nach rechts ziehen / Groß nach oben ziehen / Groß nach unten ziehen / Groß nach links ziehen / Groß nach rechts ziehen] Auswahl des Übergangs-Effekts zum Ändern der Seite in der Visualisierung.

PC/Tablet-Auto/Hochskalierung Größe: Aktivierung der automatischen Re-Skalierung für mehrere Bildschirmauflösungen.

PC/Tablet Hintergrundfarbe: Gemeinsame Hintergrundfarbe für die Visualisierung.

PC/Tablet Hintergrundbild: Gemeinsames Hintergrundbild für die Visualisierung.

Benutzerdefinierte Schriftart: Gemeinsame Schriftart für die Visualisierung.

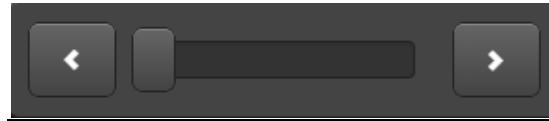
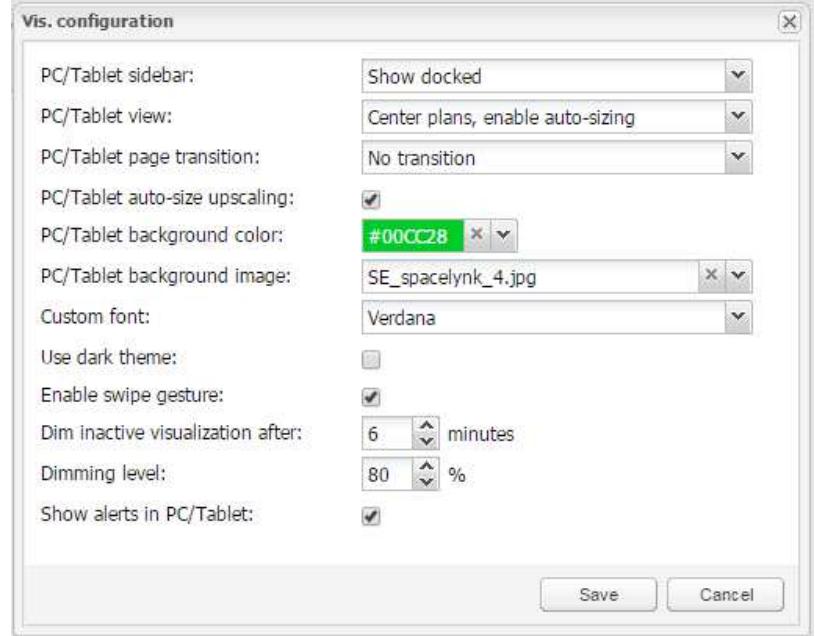
Dunkles Design verwenden: Umkehrung von Farben, Schriftarten und Steuerungen, um Visualisierungen dunkler Paletten zu entsprechen.

Inaktive Visualisierung dimmen nach: ? Minuten: Funktion zur Energieeinsparung bei batteriebetriebenen Geräten.

Inaktive Visualisierung dimmen nach: ? Minuten: Funktion zur Energieeinsparung bei batteriebetriebenen Geräten.

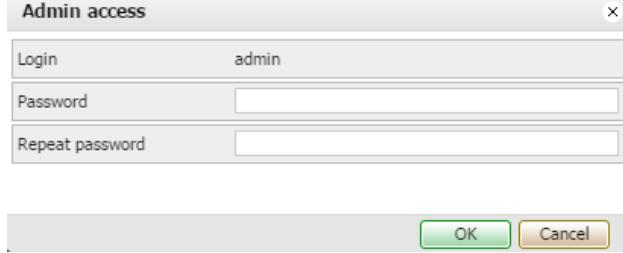
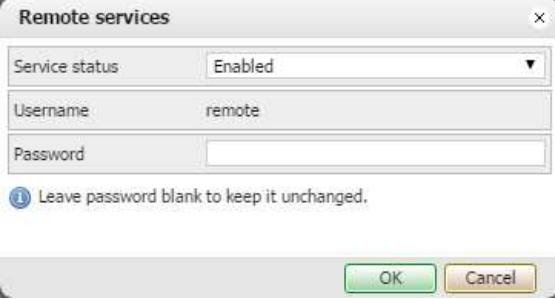
Dimmwert ?%: Helligkeitswert gedimmter Bildschirme.

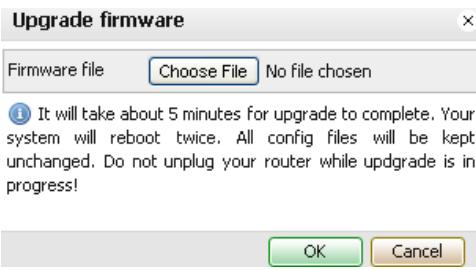
Warnungen in PC/Tablet anzeigen: Nach Auslösung einer neuen Warnung wird dies in der Visualisierung von PC/Tablet angezeigt.



3.3 System

3.3.1 System → System

<p><u>System</u></p> <p>Erlaubt die Verwaltung der Router-Funktionalität im KNX/EIB spaceLYnk und den Zugriff auf Steuerungsmanagement und Firmware-Upgrade.</p>	
<p><u>KNX-Verbindung</u> Siehe Kapitel 3.3.2 für Details.</p>	
<p><u>Administratorzugriff</u></p> <p>Kennworteinstellungen für das Administrator-Konto. Der Standard-Benutzername ist admin. Die Konfiguration für Anmeldung und Kennwort für den Benutzerzugriff befindet sich im Hauptmenü.</p>	
<p><u>Remotedienste</u></p> <p>Zur Aktivierung/Deaktivierung des Remote-Zugriffs auf spaceLYnk für Servicezwecke.</p>	

<p><u>Upgrade Firmware</u></p> <p>System → Upgrade firmware wird zur Durchführung eines vollständigen Upgrade des Systems verwendet (Sowohl OS als auch der spaceLYnk-Teil).</p> <p>HINWEIS: Es wird nach jedem Upgrade empfohlen, den Browser-Cache zu leeren.</p> <p>Während des Upgrade der Firmware reagiert das Gerät nicht, da spaceLYnk mehrmals neu gestartet wird.</p>	
<p><u>Neustart</u></p> <p>Durch Ausführung des Befehls System → Neustart wird spaceLYnk neu gestartet.</p>	
<p><u>Herunterfahren</u></p> <p>Durch Ausführung des Befehls System → Herunterfahren wird spaceLYnk heruntergefahren.</p> <p>HINWEIS: Es wird dringend geraten, das System vor dem Abschalten herunterzufahren, sodass die Datenbank sicher gesichert werden kann. Das System ist heruntergefahren, wenn die LED Nr. 1 zu blinken aufhört. Die einzige Art, spaceLYnk wieder auf AN zu stellen, ist das Trennen und Neuverbinden der Stromzufuhr. Es kann nicht Remote ANGESTELLT werden.</p>	

3.3.2 System → Netzwerk

Schnittstellen

Die Schnittstelle Ethernet wird in der ersten Registerkarte aufgelistet. Das Flussdiagramm für den Datenverkehr kann  mit Verwendung der Schaltfläche Diagramm an der rechten Seite geöffnet werden.

Durch Klicken auf die Benutzeroberfläche erscheint das Fenster Konfiguration.

Protokoll – Spezifisches, zur Adressierung verwendetes Protokoll.

- **Keines** – Es wird kein Protokoll verwendet.
- **Statische IP** – Statische IP-Adresse. Standardmäßig **192.168.0.10**.
- **DHCP** – Verwenden Sie das DHCP-Protokoll zum Abrufen der IP-Konfiguration.
- **Aktuelle IP** – Die vom DHCP-Server erhaltene IP-Adresse. Dieses Feld erscheint nur, wenn eine IP-Adresse vergeben wurde. Es bleibt ansonsten verborgen.

IP-Adresse – Standardmäßig **192.168.0.10**.

Netzwerkmaske – Netzwerkmaske. Standardmäßig **255.255.255.0** (/24).

Gateway IP – Gateway-IP-Adresse.

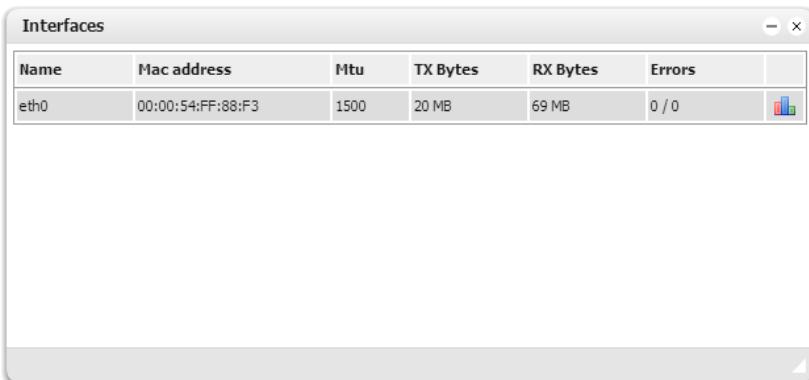
DNS server 1 – Primäre IP-Adresse des DNS-Servers.

DNS server 2 – Sekundäre IP-Adresse des DNS-Servers.

Mtu – Maximale Übertragungseinheit. Die maximale Größe des Pakets, das im Kommunikationsprotokoll übergeben werden kann. (Standard 1500).

Daten der Ethernet-Schnittstelle über Grafik – Wenn Sie im Hauptfenster der Registerkarte

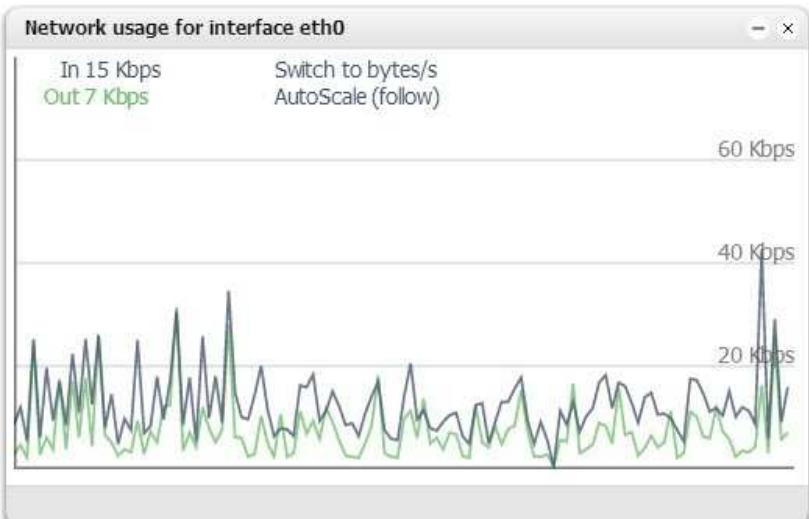
Ethernet auf die  Schaltfläche klicken, wird ein neues Fenster geöffnet. Zeichnet eine Echtzeit-Grafik des Datenverkehr-Flusses, der die Schnittstelle passiert (sowohl eingehend, wie auch ausgehend). Die Maßeinheiten können geändert werden – Bits/s oder Bytes/s und Grafik Automatische Skalierung oder Nach oben.



Interface eth0

Protocol	Static IP
IP address	10.154.20.25
Network mask	255.255.255.0
Gateway IP	10.154.20.1
DNS server 1	10.154.20.1
DNS server 2	
MTU	

OK **Cancel**



Routen

Routingtabelle ist eine Datentabelle, welche die Routen zu einem bestimmten Netzwerkziel auflistet. Sie beinhaltet Informationen über die Topologie unmittelbar um sie. Die Routingtabelle des Systems befindet sich im **Menü Netzwerk→routen**. Das Fenster ist in zwei Teile unterteilt – **Dynamische** und **Statische** Routen.

- Dynamisch**

Liste aller selbst erlernten Netzwerkziele und automatische Auswahl der „Besten Route“.

Schnittstelle – Der Schnittstellenname zeigt die lokal verfügbare Schnittstelle an, die für das Erreichen des Gateways verantwortlich ist.

Ziel – Die Ziel-Subnetz-IP-Adresse beschreibt zusammen mit der Netzwerkmaske die Netzwerk-ID.

Gateway – Gateway-IP-Adresspunkte zum Gateway, durch die das Netzwerk erreicht werden kann.

Netzwerkmaske – Netzwerkmaske.

Kennzeichen – Hilft bei der Fehlerbehebung Ihres Netzwerkproblems. Sehen Sie dazu die beigelegte Kodierungstabelle.

- Statisch**

Manuelle Eingabe der Routen, die sich nicht automatisch ändern, in die spaceLYnk-Routingtabelle .

Schnittstelle– Schnittstellenname.

Ziel – Ziel-IP-Adresse.

Gateway – Gateway-IP-Adresse.

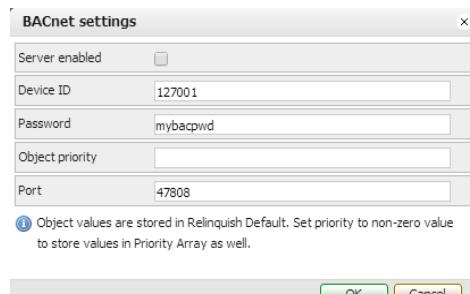
Netzwerkmaske – Netzwerkmaske.

Kennzeichen – Hilft bei der Fehlerbehebung Ihres Netzwerkproblems. Sehen Sie dazu die beigelegte Kodierungstabelle.

Routes				
Interface	Destination	Gateway	Network mask	Flags
eth0	default	10.154.20.1	0.0.0.0	UG
eth0	10.154.20.0	*	255.255.255.0	U
eth0	224.0.0.0	*	224.0.0.0	U

Kennzeichen	Name	Bedeutung
1	RTF_PROTO1	Protokollspezifisches Routing Kennzeichnung 1
2	RTF_PROTO2	Protokollspezifisches Routing Kennzeichnung 2
3	RTF_PROTO3	Protokollspezifisches Routing Kennzeichnung 3
B	RTF_BLACKHOLE	Einfach Pakete verwerfen (während Aktualisierungen)
b	RTF_BROADCAST	Diese Route stellt eine Broadcastadresse dar
C	RTF_CLONING	Neue Routen bei Verwendung erstellen
c	RTF_PRCLONING	Protokollspezifisch neue Routen bei Verwendung erstellen
D	RTF_DYNAMIC	Dynamisch erstellt (durch Umleiten)
G	RTF_GATEWAY	Ziel erfordert Weiterleitung durch Zwischenstufe
H	RTF_HOST	Hosteintrag (ansonsten Net)
L	RTF_LLINFO	Gültiges Protokoll zur Verknüpfung der Adressübersetzung
M	RTF_MODIFIED	Dynamisch geändert (durch Umleiten)
R	RTF_REJECT	Host oder Net nicht erreichbar
S	RTF_STATIC	Manuell hinzugefügt
U	RTF_UP	Route verwendbar
W	RTF_WASCLONED	Route wurde als Ergebnis von Klonen erstellt
X	RTF_XRESOLVE	Externer Daemon übersetzt Proto zu Link-Adresse

BACnet-Einstellungen



BACnet-Objekte

Type	Instance	Device name	Current value
2 (AV)	1	CO2 (0.0.1)	412.8
2 (AV)	2	Humidity (0.0.2)	17
2 (AV)	3	Temperature (0.0.3)	22.08
2 (AV)	5	Maximal CO2 (0.0.5)	522.88
2 (AV)	7	Maximal humidity (0.0.7)	40
5 (BV)	256	Window 1 (0.1.0)	false
5 (BV)	257	Window 2 (0.1.1)	false
5 (BV)	258	Window 3 (0.1.2)	false
5 (BV)	270	Central OFF (0.1.14)	false
5 (BV)	512	Office 1 PIR (0.2.0)	true
5 (BV)	513	Office 2 PIR (0.2.1)	true
2 (AV)	1635	Total current (0.6.99)	9.73
2 (AV)	1792	Consumption relay 1 M1 (0.7.0)	17965
2 (AV)	1793	Consumption relay 2 W1 (0.7.1)	16783
2 (AV)	1794	Consumption relay 3 W2 (0.7.2)	15154
2 (AV)	1892	Price (0.7.100)	178.08
2 (AV)	1904	percents office 2 (0.7.112)	99
2 (AV)	2048	Temperature heating (1.0.0)	24

HINWEIS: Siehe Kapitel 5 für weitere Details.

ARP-Tabelle

Die Adressenauflösungs-Protokoll-Tabelle ist in der **Netzwerk → -APR-Tabelle aufgelistet.**

Wird zur Auflösung von Netzwerk-Layer-Adressen in Link-Layer-Adressen verwendet.
Wandelt eine IP-Adresse in eine physikalische Adresse um.

Interface	IP address	Mask	MAC address	Flags
eth0	10.154.16.243	*	2c:54:2d:00:da:d0	0x2
eth0	10.154.16.248	*	00:0e:b6:a0:68:a2	0x2
eth0	10.154.20.1	*	00:07:7d:a7:e8:2e	0x2

KNX-Verbindung

Die KNX-spezifische Konfiguration befindet sich in **Konfigurator -> Dienstprogramme -> Netzwerk** → **Netzwerk** → **Fenster KNX-Verbindung**.

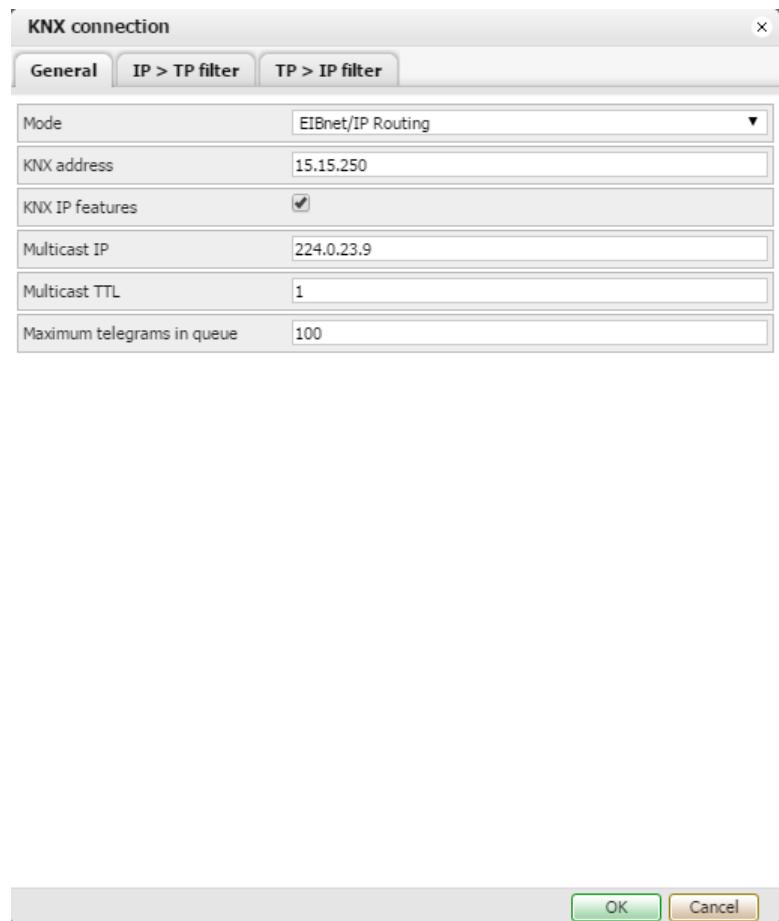
Allgemein

Modus- KNX-Verbindungsmodus. spaceLYnk verfügt standardmäßig über eine eingebaute TP-UART-Schnittstelle.

- **TP-UART**– Twisted Pair-Verbindung über schwarz/roten Stecker.
Übertragungsrate 9.6 kB/s.
- **EIBnet/ IP Tunneling**– IP-Verbindung, maximal 1000x schneller als TP-UART. spaceLYnk als Server. Unicast, anerkannter Datenaustausch, zusätzlich individuelle Adresse über Tunneling - Verbindung.
- **EIBnet/ IP Tunneling (NAT mode) – Netzwerkadressübersetzung-Modus**– Ermöglicht die Verbindung mehrerer Geräte mit einem öffentlichen Netzwerk und Verwendung der gleichen, öffentlichen IPv4-Adresse. Ändert während der Übertragung über ein Datenverkehr-Routing-Gerät die IP-Adressinformation in den IPv4-Headers.
- **EIBnet/IP Routing** – Multicast, nicht bestätigter Datentransfer. spaceLYnk verfügt über einen Line- oder Backbone-Coupler.
- **ACK aller Gruppentelegramme** – spaceLYnk muss die empfangenen Telegramme bestätigen, wenn es direkt mit einem anderen KNX-Gerät kommuniziert. Auswahl aufheben, wenn spaceLYnk nur als Sniffer von Gruppenadressen arbeitet.

KNX Adresse – KNX-physikalische Adresse des Geräts.

KNX IP Funktionen – Verwenden Sie dieses Gerät mit KNX IP-Funktionen, beispielsweise



<p>KNXnet/IP-Netzwerkkonfiguration. Wenn inaktiv, werden alle IP-Kommunikationen von KNX geblockt.</p> <p>Multicast IP – Multicast-IP-Adresse.</p> <p>Multicast TTL – Standardwert ist 1. Ermöglicht die Kommunikation zwischen unterschiedlichen Subnetzwerken.</p> <p>Maximale Telegramme in Warteschlange – Anzahl der maximalen Telegramme in der Warteschlange.</p> <p>HINWEIS: Wenn der KNX TP nicht mit einem Gerät verbunden ist, sollte der Routing-Modus verwendet werden, um sicherzustellen, dass die Gruppenadressen korrekt aktualisiert werden. Nach Anwendung der Änderungen startet das Gerät neu.</p>	
---	--

IP > TP-Filter

Filter akzeptiert oder verwirft Telegramme von den definierten KNX-Geräten/physikalischen Adressen. Ausgehende Telegramme werden nicht gefiltert.

Filter zu Tunneling anwenden – Dieser Filter wurde entwickelt, um eine erweiterte Funktion im Vergleich zu einem Standard KNX-Router zu bieten. Der definierte Filter kann sogar jetzt für den Tunneling-Modus angewendet werden. Standardmäßig durchläuft er alle Telegramme. Diese Option gilt für beide Richtungen (IP > TP-Filter & TP > IP-Filter)

SRC Richtlinie [Kein Filter / Annehmen ausgewählter, individueller Adressen / Verweigern ausgewählter, individueller Adressen] – Richtlinie zur Anwendung bei der Liste von Quelladressen.

Ind. Adressliste – Listet individuelle Adressen oder Gruppenadressen. Eine Adresse pro Zeile. Verwenden * (z.B. 1.1.* oder 1/1/*), um alle Adressen in der bestimmten Zeile zu filtern.

DST-Gruppenrichtlinie

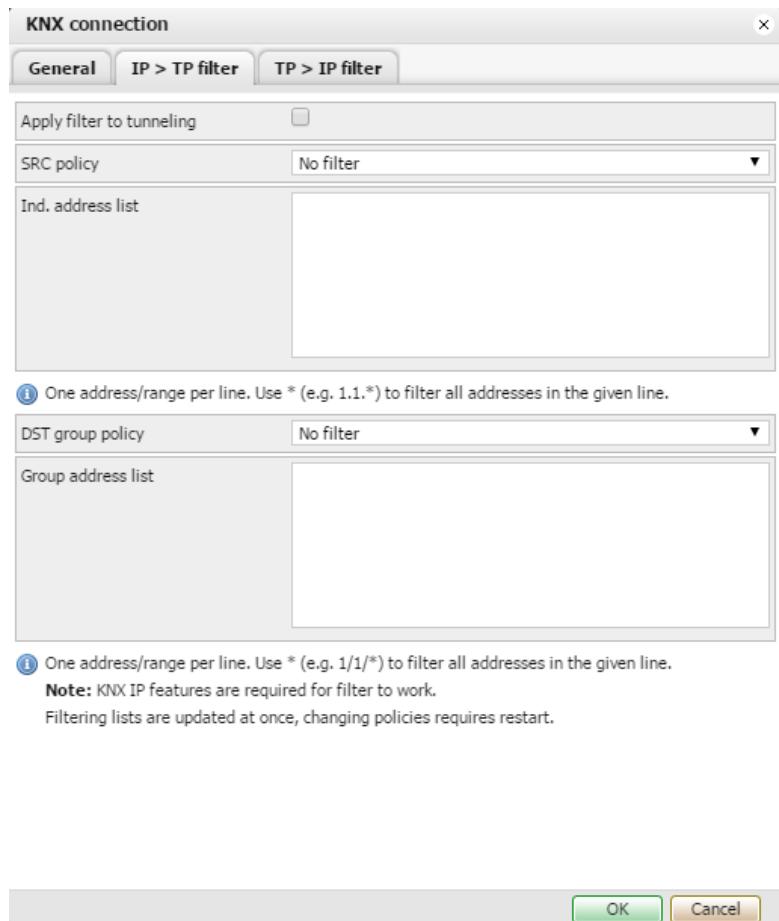
Filter der Zielgruppe akzeptiert oder verwirft empfangene Telegramme, die zu einer Gruppe als 1/2/3, oder einer Untergruppe als 1/2/* gehören. Ausgehende Telegramme werden nicht gefiltert.

DST-Gruppenfilter [Kein Filter / Annehmen ausgewählter, individueller Adressen / Verweigern ausgewählter, individueller Adressen] – Richtlinie zur Anwendung bei der Liste von Zielgruppen-Adressen.

Gruppenadress-Liste – Liste der Gruppenadressen.

Eine Adresse pro Zeile. Verwenden * (z.B. 1/1/*), um alle Adressen in der bestimmten Zeile zu filtern.

HINWEIS: KNX IP-Funktionen sollten aktiviert werden, damit die Filter funktionieren. Dies gilt nur für eingehende Telegramme!



TP > IP-Filter

Filter akzeptiert oder verwirft Telegramme von den definierten KNX-Geräten/physikalischen Adressen. All die ausgehenden Telegramme werden nicht gefiltert.

Filter an virtuellen Objekten anwenden –

Virtuelle Objekte werden zum internen Datenaustausch innerhalb von spaceLYnk verwendet (z.B. Modbus zu Visu). Wenn der Befehl gr.update() in Lua verwendet wird, So wird die Gruppenadresse nicht auf TP, sondern nur auf IP geschrieben. Wenn diese Option markiert ist werden die gelisteten Gruppen von der IP, und somit virtuell, gefiltert (=nicht geschrieben).

SRC Richtlinie [Kein Filter / Annehmen ausgewählter, individueller Adressen / Verweigern ausgewählter, individueller Adressen] – Richtlinie zur Anwendung bei der Liste von Quelladressen.

Ind. Adressliste – Liste der individuelle Adressen oder Gruppenadressen. Eine Adresse pro Zeile. Verwenden * (z.B. 1.1.* oder 1/1/*), um alle Adressen in der bestimmten Zeile zu filtern.

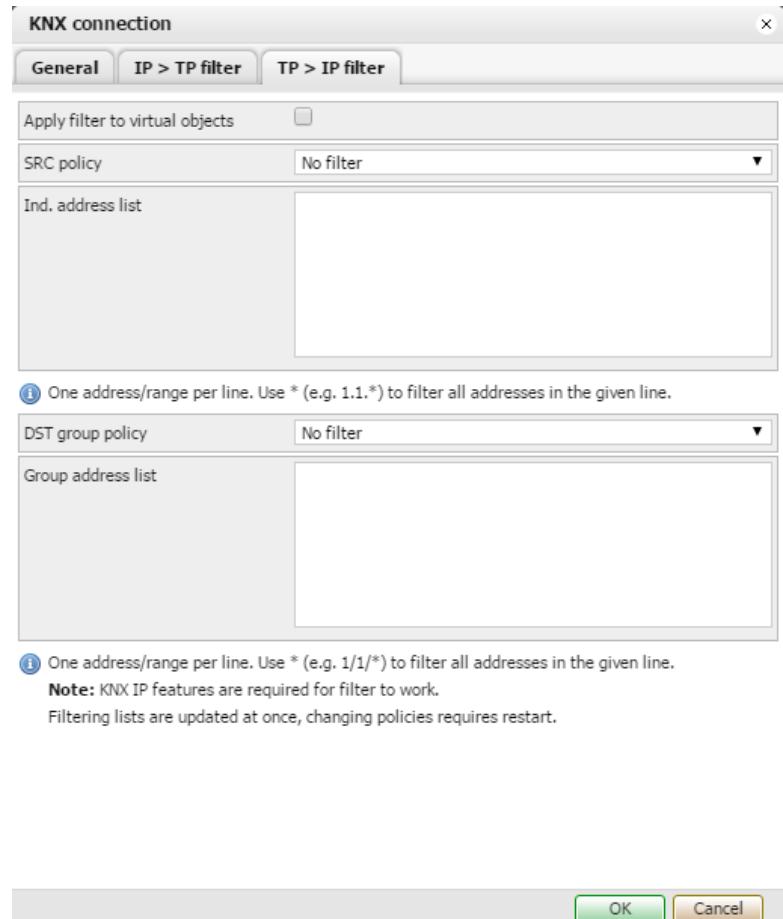
DST-Gruppenrichtlinie – Filter der Zielgruppe akzeptiert oder verwirft empfangene Telegramme, die zu einer Gruppe als 1/2/3, oder einer Untergruppe als 1/2/* gehören.
Ausgehende Telegramme werden nicht gefiltert.

DST-Gruppenfilter [Kein Filter / Annehmen ausgewählter, individueller Adressen / Verweigern ausgewählter, individueller Adressen] – Richtlinie zur Anwendung bei der Liste von Zielgruppen-Adressen.

Gruppenadress-Liste – Liste der Gruppenadressen.

Eine Adresse pro Zeile. Verwenden * (z.B. 1/1/*), um alle Adressen in der bestimmten Zeile zu filtern.

HINWEIS: KNX IP-Funktionen sollten aktiviert werden, damit die Filter funktionieren. Dies gilt nur für die eingehenden Telegramme!



3.3.3 System → Dienste

<p>NTP-Client Network Time Protocol (Synchronisation der Uhr) Server 1- 4. Definieren Sie den Server, von dem Zeit und Datum erhalten wird. HINWEIS: Neustart erforderlich.</p>	
<p>FTP-Server Durch Aktivierung von Dienst FTP-Server kann auf den FTP- → Server von spaceLYnk zugegriffen werden. Serverstatus – Sicherer Tunnelmodus. Port – Port des Dienstes. Benutzername – Benutzername, ftp. Kennwort – Kennwort , Länge 4-20 Symbol. Das Standardkennwort lautet ftp. Min. Port passiver Modus – Minimaler Port für passiven Modus. Max. Port passiver Modus – Maximaler Port für passiven Modus.</p>	<p>① Leave password to blank to keep it unchanged. External IP and passive mode ports must be set when you want to access FTP behing NAT. Make sure both FTP port and passive mode port range are forwarded on your router</p>
<p>Systemüberwachung Definition von automatische Prüfung System und automatischer Neustart.</p>	<pre># check once in 2 minutes set daemon 120 # reboot system when memory or cpu usage is too high check system \$HOST if cpu usage (user) > 90% for 15 cycles then exec "/sbin/reboot" if memory usage > 90% for 5 cycles then exec "/sbin/reboot" # knx backend check process eibd with pidfile /var/run/eibd start program = "/etc/init.d/eibd restart" stop program = "/etc/init.d/eibd stop" if 5 restarts within 5 cycles then timeout # knx monitor check process groupmonitor with pidfile /var/run/gs-groupmonitor.pid start program = "/etc/init.d/genohm-scada restart" stop program = "/etc/init.d/genohm-scada stop" if 5 restarts within 5 cycles then timeout</pre>

3.3.4 System → Status

Systemstatus

Die Systeminformation wird in den folgenden Registerkarten angezeigt:

System status	
General	
Parameter	Value
CPU model	ARM926EJ-S rev 5 (v5l)
CPU BogoMips	226.09
Linux kernel version	3.10.30
System uptime	5d 22h 27m
Load averages	0.09 0.19 0.30

Allgemein

Hardwareinformation, vom Kernel bereitgestellt.

Specherauslastung

Aktuell vom System verwendeter Speicher.

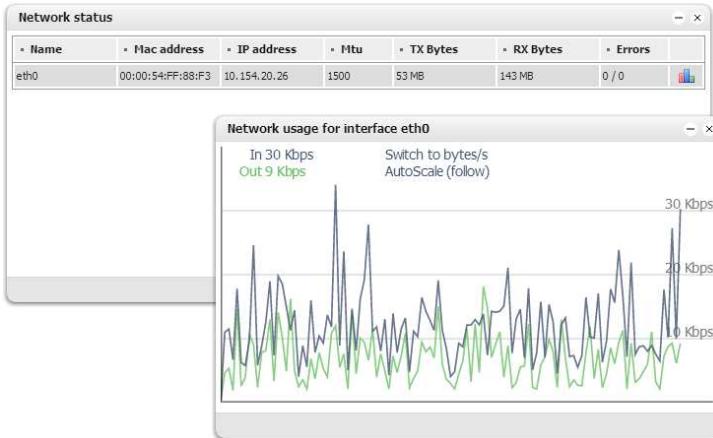
Partitionen

Liste der im System verfügbaren Partitionen.

Serielle Ports

Liste der im System verfügbaren seriellen Ports.

Netzwerkstatus



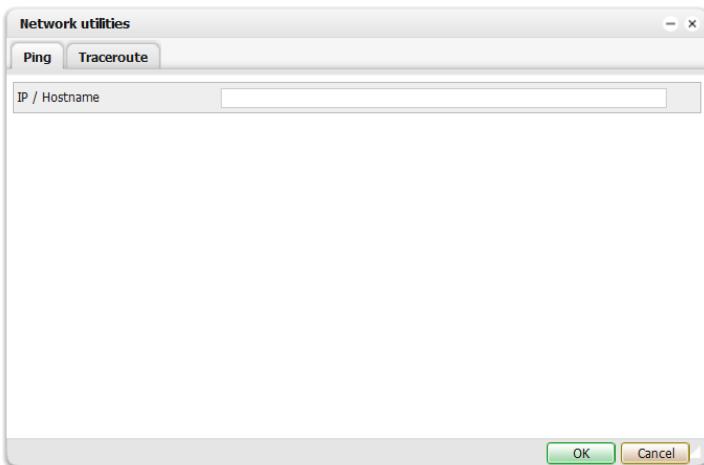
Netzwerk-Dienstprogramme

Ping

Die Computer Netzwerktool-ID, die zum Test verwendet wird, ob ein bestimmter Host über ein IP-Netzwerk erreicht werden kann.

Ablaufverfolgung Route

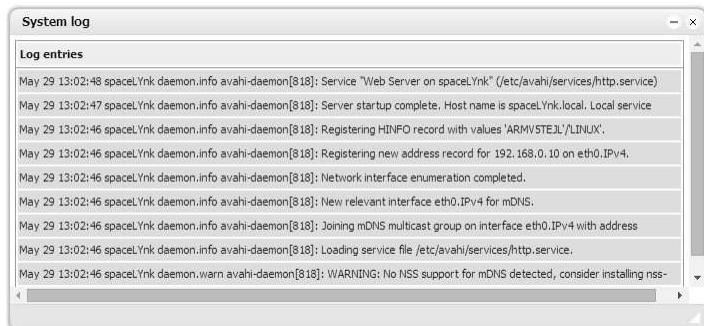
Das Computer Netzwerk-Diagnosetool wird zur Anzeige der Route (Pfad) und zur Messung der Übergabeverzögerung von Paketen über ein Internetprotokoll (IP)-Netzwerk verwendet.



Systemprotokoll

Protokolleinträge

spaceLYnk erstellt und verwaltet Protokolldateien über alle durchgeführten Aktivitäten automatisch.



Ausgeführte Prozesse

Running processes	
PID	Command
1	[init]
2	[kthreadd]
3	[ksoftirqd/0]
4	[kworker/0:0]
5	[kworker/0:0H]
6	[kworker/u:0]
7	[kworker/u:0H]
8	[rcu_preempt]
9	[rcu_bh]
10	[rcu_sched]
11	[watchdog/0]
12	[khelper]
13	[kdevtmpfs]
14	[kworker/u:1]
141	[bdl-default]
142	[integrityd]

3.4 Objekte

List der KNX-Netzwerkobjekte erscheint im Menü **Objekte**. Die Objekte werden folgendermaßen gelistet:

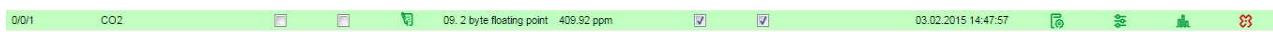
1. Durch Sniffing des Bus für Telegramme von unbekannten Gruppenadressen aufgezeichnet (wenn in **Dienstprogrammen** aktiviert).
2. Manuell hinzugefügt.
3. Importieren einer ESF-Datei (in **Dienstprogramme**).

Objekte werden mit den folgenden Parametern sortiert– **Gruppenadresse, Objektname, IP>TP-Filter, TP>IP-Filter, Ereignisskript, Datentyp, Aktueller Wert, Protokoll, Export, Tags, Aktualisiert am, Wert festgelegt, Vis.parameters und Benutzerdefinierte Werte.**

Object filter	Group address	Object name	IP > TP fil...	TP > IP fil...	Event sc...	Data type	Current value	Log	Export	Tags	Updated at	Set value	Vis. para...	Custom ...
Name or group address:	0/01	CO2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09. 2 byte floating point	409.92 ppm	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		03.02.2015 14:45:57	<input type="button"/>	<input type="button"/>	<input type="button"/>
	0/02	Humidity	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	05.001 scale	21 % RH	<input checked="" type="checkbox"/>	<input type="checkbox"/>		03.02.2015 14:45:20	<input type="button"/>	<input type="button"/>	<input type="button"/>
Data type:	0/03	Temperature	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09.001 Temperature	23.14 °C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		03.02.2015 14:45:13	<input type="button"/>	<input type="button"/>	<input type="button"/>
Not specified	0/04	Minimal CO2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09. 2 byte floating point	341.78 ppm	<input checked="" type="checkbox"/>	<input type="checkbox"/>		31.01.2015 13:55:42	<input type="button"/>	<input type="button"/>	<input type="button"/>
Tags:	0/05	Maximal CO2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09. 2 byte floating point	442.88 ppm	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		22.01.2015 10:16:48	<input type="button"/>	<input type="button"/>	<input type="button"/>
	0/06	Minimal humidity	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09. 2 byte floating point	16 %RH	<input checked="" type="checkbox"/>	<input type="checkbox"/>		26.01.2015 14:27:33	<input type="button"/>	<input type="button"/>	<input type="button"/>
	0/07	Maximal humidity	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09. 2 byte floating point	40 %RH	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		26.01.2015 10:05:25	<input type="button"/>	<input type="button"/>	<input type="button"/>
Match mode:	0/08	Minimal temperature	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09.001 Temperature	22.18 °C	<input checked="" type="checkbox"/>	<input type="checkbox"/>		02.02.2015 05:50:12	<input type="button"/>	<input type="button"/>	<input type="button"/>
All tags	0/09	Maximal temperature	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09.001 Temperature	28.78 °C	<input checked="" type="checkbox"/>	<input type="checkbox"/>		03.02.2015 10:27:25	<input type="button"/>	<input type="button"/>	<input type="button"/>
Any tag	0/10	Climate values reset	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	09. 2 byte floating point	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>		20.01.2015 15:25:57	<input type="button"/>	<input type="button"/>	<input type="button"/>
	0/11	Climate time	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	10. 3 byte time / day	15:25:58, Tuesday	<input checked="" type="checkbox"/>	<input type="checkbox"/>		20.01.2015 15:25:58	<input type="button"/>	<input type="button"/>	<input type="button"/>
	0/12	Climate date	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	11. 3 byte date	20.01.2015	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		20.01.2015 15:25:58	<input type="button"/>	<input type="button"/>	<input type="button"/>

Für einen schnellen Überblick werden Objekte außerdem durch deren Hintergrundfarbe unterschieden:

- Grün – Objektwert tatsächlich aktualisiert.



- Gelb – Objekt tatsächlich von einer Bus-Sniffer erkannt.



3.4.1 Objektparameter

Objektname – Name für das Objekt.

Gruppenadresse – Gruppenadresse für dieses Objekt.

Datentyp – KNX-Datentyp für das Objekt. Dies muss festgelegt werden, wenn spaceLYnk das neue Objekt zur eigentlichen Arbeit snifft.

Aktueller Wert– Tatsächlicher Wert des Objekts.

Tags– Weisen Sie dieses Objekt einigen Tags zu, die später in den Schreiberkripten, beispielsweise *Alle_Lichter_Erster_Stock* verwendet werden können. (Für Anwendungsfälle beziehen Sie sich bitte auf die Skript-Bibliothek).

Einheit/suffix– Fügen Sie dem Wert eines Objekts eine Einheit/Suffix hinzu. Sie können Einheiten, die nicht von der Tastatur aus erstellt werden können, mit einem externen Editor oder durch einfügen in den Browser erstellen.

Protokoll– Protokollierung für dieses Objekt aktivieren. Protokolle erscheinen in der Registerkarte *Objektprotokolle*.

Protokoll mit hoher Priorität – Diese Option verschiebt Protokolle mit hoher Priorität in der Bildschirmliste (Registerkarte Protokolle). Wenn die für Protokolle definierte Begrenzung überschritten wird werden Protokolle mit niedriger Priorität am Ende der Liste zuerst gelöscht. Diese Funktion stellt sicher, dass Protokolle mit hoher Wichtigkeit länger sichtbar bleiben.

Export– Macht Objekte durch Remote-SML-Anforderungen sichtbar.

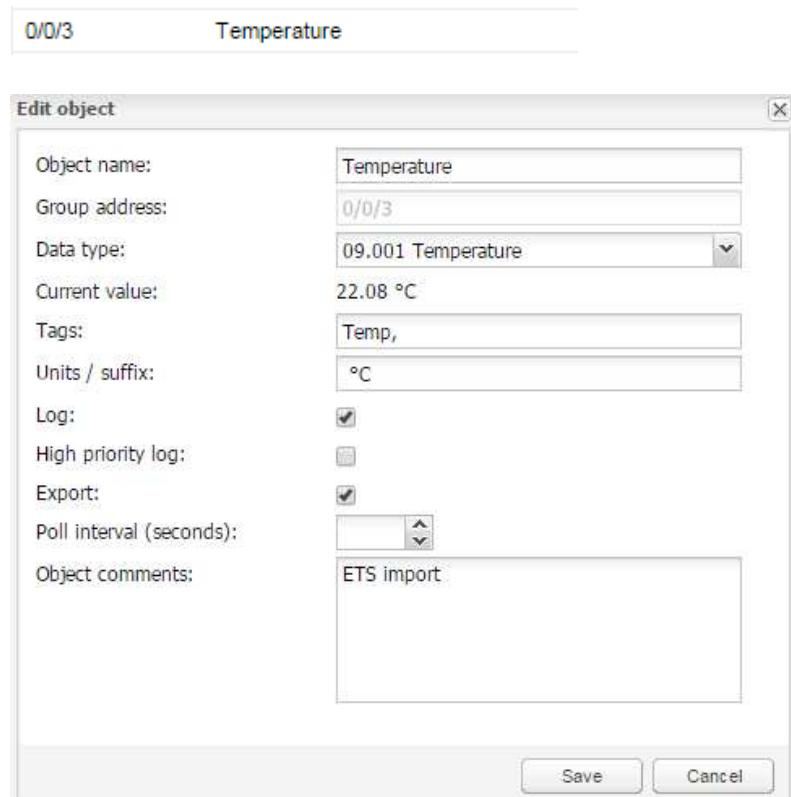
Abfrageintervall (Sekunden) – Führt nach einem ausgewählten Zeitintervall eine automatische Objektauslesung durch.

Objektbemerkungen–Weitere Beschreibung des Objekts. Kann auch zum Filtern verwendet werden.

3.4.2 Ereignisskript

Durch Drücken  auf diese Schaltfläche in der Objektliste öffnet sich der Skript-Editor, und ein Skript auf Ereignisbasis kann erstellt werden. Das Skript wird jedes Mal ausgeführt, wenn ein Telegramm an die ausgewählte Gruppe gesendet wird. Wenn das Skript an eine Gruppe angehängt wird, ändert sich das Symbol zu grün .

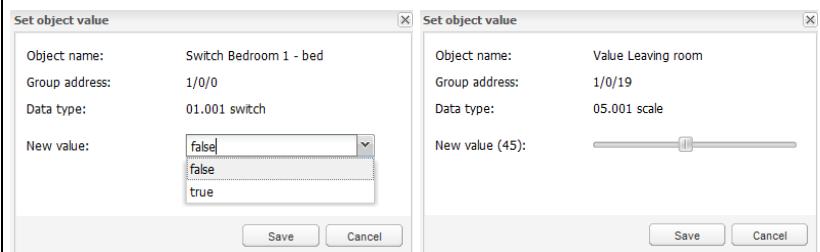
Zur Änderung der Einstellungen für existierende oder neue Objekte klicken Sie auf die Adresse oder den Namen des spezifischen Listeneintrags.



3.4.3 Objektwert festlegen

Durch Drücken auf die  Schaltfläche in der Objektliste kann der Status des Objekts geändert werden.

Das Erscheinungsbild des **Fensters Neuer Wert** hängt von den für das spezifische Objekt gesetzten Parametern zur Visualisierung ab.



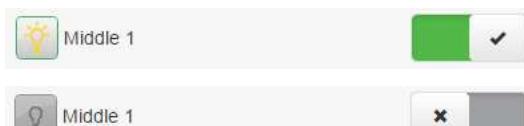
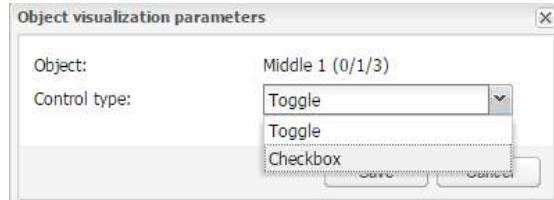
3.4.4 Parameter zur Objektvisualisierung

Durch Drücken auf diese Schaltfläche können  für das entsprechende Objekt spezifische Parameter zur Visualisierung für diesen Typ gesetzt werden.

1-Bit

Steuerungstyp – Typen des visuellen Steuerelements:

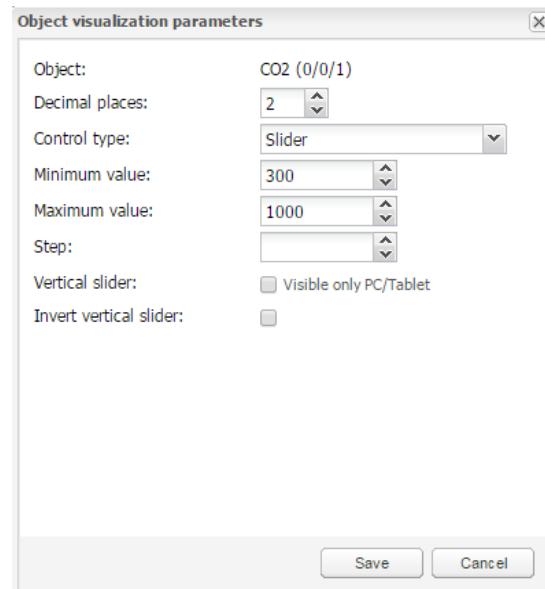
- **Umschalten**
- **Kontrollkästchen**



4-Bit (3-Bit gesteuert)

Schrittgröße – Beispiel Schrittgröße zur Steuerung der Rollläden:

2-Bit (1-Bit gesteuert), 1-Byte-Ganzzahl ohne Vorzeichen (Skala), 1-Byte-Ganzzahl mit Vorzeichen, 2-Byte-Ganzzahl ohne Vorzeichen, 2-Byte-Ganzzahl mit Vorzeichen, 2-Byte Fließkomma (Temperatur), 4-Byte-Ganzzahl ohne Vorzeichen, 4-Byte-Ganzzahl mit Vorzeichen, 4-Byte Fließkomma.



Steuerungstyp – Typen des visuellen Elements:

Direkt +/-

Schieber

Kreisschieber

Benutzerdefinierte Wertauswahl

Minimalwert – Definieren Sie einen Minimalwert nur zur Visualisierung.

Maximalwert – Definieren Sie einen

Maximalwert nur zur Visualisierung.

Schritt – Wenn definiert ändert sich der Wert abhängig vom definierten Schritt.

Vertikaler Schieber – Vertikale Position des Schiebers, für beispielsweise die Steuerung der Rollläden.

Umkehrung vertikaler Schieber – Kehrt die Richtung des vertikalen Schiebers um.

Farbe des Schiebers – Definiert die Füllfarbe des Schiebers.

Hintergrundfarbe – Definiert die Hintergrundfarbe des Kreisschiebers.

Kappe runde Linie – Rundungskanten des Kreisschiebers.

Titel verbergen –

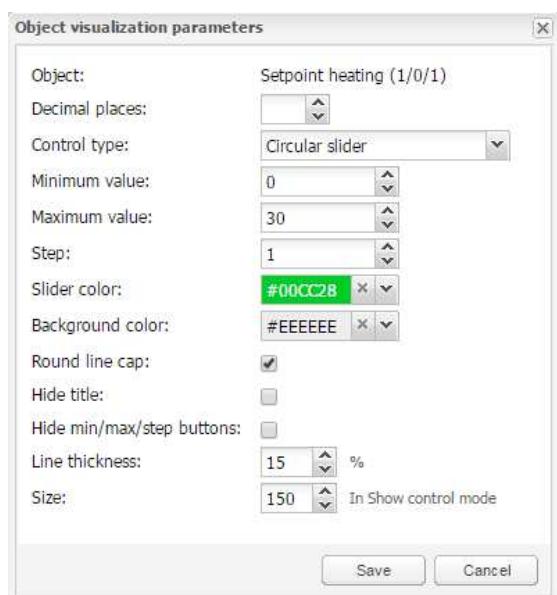
Objektnamen/benutzerdefinierten Namen in der Visualisierung verbergen.

Schaltflächen Min/Max/Schritt verbergen –

Schaltflächen in der Visualisierung für Touchscreen-fähige Geräte verbergen

Liniенstärke – Stärke der Steuerlinie Kreisschieber (1-50%).

Größe – Größe des Kreisschiebers (150-500 Pixel).



3.4.5 Benutzerdefinierter Textwert

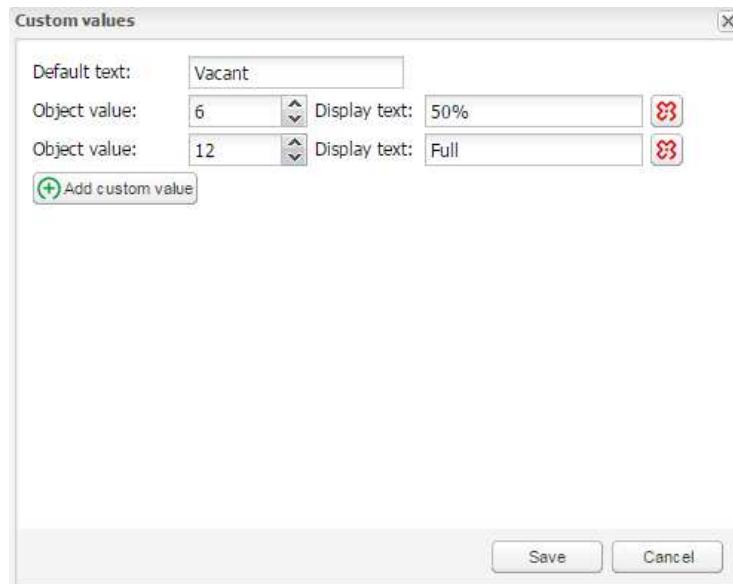
Durch Drücken dieser Schaltfläche  kann der benutzerdefinierte Text in der Objektliste den Objektwerten hinzugefügt werden.

Benutzerdefinierte Textwerte können nur auf Boolesch oder ganzzahlige Werte eingestellt werden.

Standardtext – Angezeigter Text, wenn kein Wert definiert ist.

Objektwert – Fügen Sie einen

benutzerdefinierten Wert hinzu, wählen Sie einen **Objektwert** aus und **definieren** Sie einen Anzeigetext.



Parking spaces



Beispiel eines angezeigten Werts mit den obigen Einstellungen in der Visualisierung.

3.4.6 Objekt-Steuerleiste

Neues Objekt hinzufügen – Fügen Sie der Liste manuell ein neues Objekt hinzu.

Automatische Aktualisierung aktiviert – Spezifiziert, ob die Objektliste automatisch aktualisiert wird.

Löschen – Löscht die Liste der gefilterten Gruppenadressen.

Massenbearbeitung – Zur Massenbearbeitung der Filter ausgewählter Objekte.

Nächste/Vorherige Seite – Zur nächsten oder vorherigen Seite wechseln.

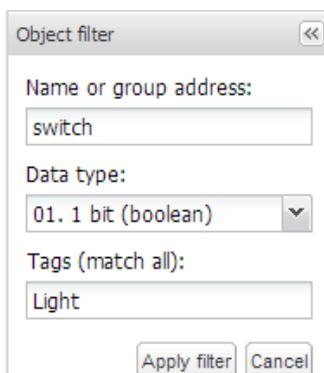
Aktualisieren – Aktualisieren der Objektliste.

TP>IP Richtlinie – Richtlinie zur Filterung auswählen



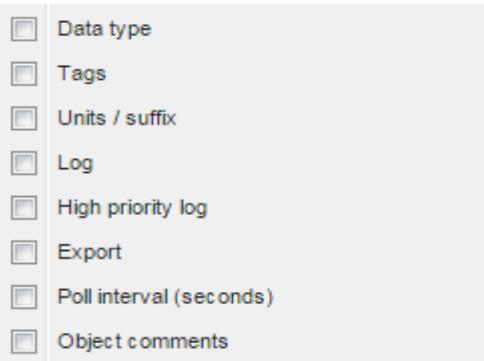
3.4.7 Objektfilter

An der linken Seite der Objektliste können Sie filtern. Geben Sie zur Anwendung des Objektfilters Namen oder Gruppenadresse ein, markieren oder spezifizieren den Datentyp des Objekts und drücken dann auf die Schaltfläche **Filter anwenden**.



3.4.8 Massenbearbeitung

Im **Objektfilter** gefilterte Objekte können massenbearbeitet werden durch



Object filter						
Group address	Object name	IP > TP fl...	TP > IP fl...	Event sc...	Data type	
1/1/4	Phase 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	01.001 switch	
1/1/5	Phase 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	01.001 switch	
1/1/6	Phase 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	01.001 switch	
1/1/7	Current protection	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	01.001 switch	
7/1/2	Light meeting room	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	01.001 switch	

3.5 Objektprotokolle

In der Registerkarte **Objektprotokolle** sind historische Telegramme von Objekten verfügbar. Nach der Aktivierung der Protokollierung für ein Objekt werden alle historischen und zukünftigen Daten

Log time	Object address	Type	Source address	Object name	Decoded value	Data type	Object data (number)
04.02.2015 08:51:16	1/7/99	write	15.15.250	OS time	8:51:16, Wednesday	10. 3 byte time / day	683310
04.02.2015 08:51:06	1/7/99	write	15.15.250	OS time	8:51:05, Wednesday	10. 3 byte time / day	683305
04.02.2015 08:51:01	1/7/99	write	15.15.250	OS time	0:00:00	10. 3 byte time / day	000000
04.02.2015 08:51:01	1/7/100	write	15.15.255	CPU temperature	80.48°C	09.001 Temperature	17DC
04.02.2015 08:50:55	1/7/99	write	15.15.250	OS time	8:50:55, Wednesday	10. 3 byte time / day	683237
04.02.2015 08:50:55	0/0/2	write	1.1.4	Humidity	20 % RH	05.001 scale	33
04.02.2015 08:50:47	0/0/3	write	1.1.4	Temperature	22.8 °C	09.001 Temperature	0C74
04.02.2015 08:50:45	1/7/99	write	15.15.250	OS time	8:50:45, Wednesday	10. 3 byte time / day	68322D
04.02.2015 08:50:35	1/7/99	write	15.15.250	OS time	8:50:35, Wednesday	10. 3 byte time / day	683223
04.02.2015 08:50:31	0/0/1	write	1.1.4	CO2	440 ppm	09. 2 byte floating point	2D5F
04.02.2015 08:50:25	1/7/99	write	15.15.250	OS time	8:50:25, Wednesday	10. 3 byte time / day	683219
04.02.2015 08:50:15	1/7/99	write	15.15.250	OS time	8:50:15, Wednesday	10. 3 byte time / day	68320F
04.02.2015 08:50:05	1/7/99	write	15.15.250	OS time	8:50:05, Wednesday	10. 3 byte time / day	683205
04.02.2015 08:50:01	1/7/99	write	15.15.250	OS time	0:00:00	10. 3 byte time / day	000000
04.02.2015 08:50:01	1/7/100	write	15.15.255	CPU temperature	80.2°C	09.001 Temperature	17D5

protokolliert.

Zum Finden von Informationen für einen spezifischen Zeitraum ist eine Filterung verfügbar:

- **Startdatum** – Startdatum und Zeit für die Protokollfilterung:
- **Enddatum** – Enddatum und Zeit für die Protokollfilterung:
- **Name oder Gruppenadresse** – Spezifischer Name oder Gruppenadresse des Objekts
- **Tags** – Gruppenobjekte mit den gleichen Tags werden gefiltert
- **Wert** – Spezifischer Objektwert
- **Quelladresse** – Spezifische Quelladresse

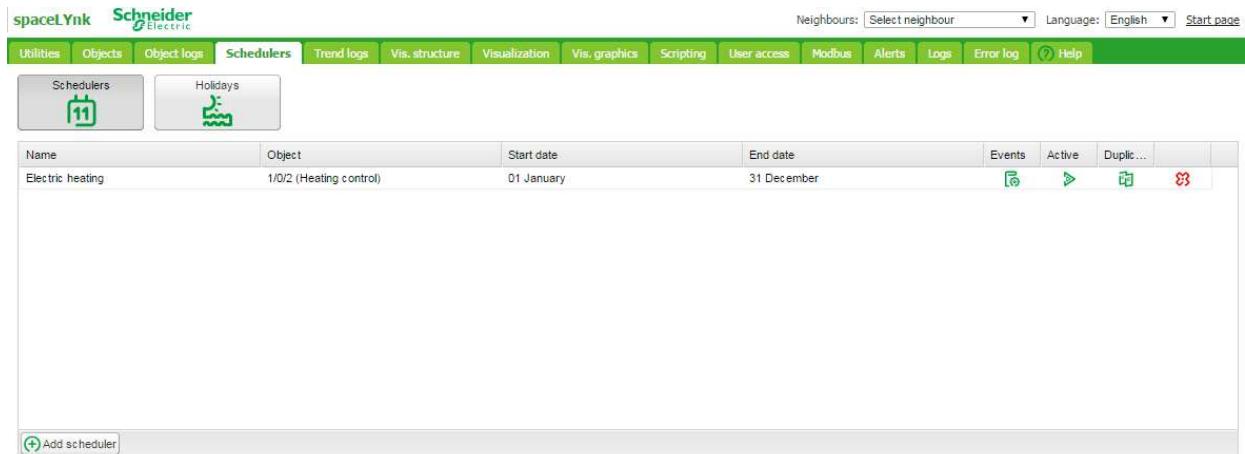
Durch Drücken auf die Schaltfläche **Löschen** können alle Protokolle gelöscht werden.

Aktivieren Sie für wichtige Objekte den Parameter Protokoll mit **Hoher Priorität** zusammen mit dem **Protokoll**-parameter. Diese Funktion listet die ausgewählten Objekte oben in der Liste der **Objektprotokolle** auf.

HINWEIS: Der Protokollspeicher wird in **Dienstprogramme → Konfigurationen festgelegt**.

3.6 Terminplaner (Schedulers)

Terminplaner ermöglichen dem Endbenutzer die Steuerung der Werte der KNX-Gruppenadressen, basierend auf Datum oder Wochentag.



3.6.1 Neuen Termin hinzufügen

Objekt– Die Objektgruppenadresse, die vom Planer gesteuert wird.

Aktiv– Definiert, ob ein Planer aktiv ist.

Name– Name des Planers.

Startdatum– Startdatum des Planers.

Enddatum– Enddatum des Planers.

Scheduler

Object:	1/0/2 Heating control
Active:	<input checked="" type="checkbox"/>
Name:	Electric heating
Start date:	01 January
End date:	31 December

Save Cancel

3.6.2 Termin Ereignisse

Ereignisse können sowohl in der Administratorenchnittstelle, oder durch den Endbenutzer in der speziellen **Benutzermodus-Planungsschnittstelle** hinzugefügt werden.

Aktiv – Definiert, ob ein Ereignis aktiv ist.

Wert – Wert, der an die Gruppenadresse gesendet wird, wenn das Ereignis ausgelöst wird.

Startzeit – Startzeit des Ereignisses.

Wochentage – Wochentage, an denen das Ereignis ausgelöst wird.

Fei – in der Registerkarte Feiertage definierte Feiertage.

Events for scheduler Electric heating				
Start time	Days of the week	Value	Active	
00:00	Mo, Tu, We, Th, Fr	20		
00:00	Sa, Su, Hol	18		
07:00	Mo, Tu, We, Th, Fr	23		
10:00	Hol	21		
18:00	Mo, Tu, We, Th, Fr	20		

Event

Active:

Value: 18

Start time: 00 00

Days of the week:

Mo Tu We Th Fr
 Sa Su Hol

Save **Cancel**

3.6.3 Feiertage

Sobald das Ereignis zur Ausführung an **Fei** maskiert ist, werden die Eingaben in Feiertag aktiviert.

HINWEIS: Der **Terminplaner**-Visualisierung wird in der **Smartphone Visualisierung** nicht unterstützt.

Holiday

Name: February

Date: 02 February 2015

Duration (days): 14

Leave year blank for recurring holidays

Save **Cancel**

3.7 Trend-Protokolle

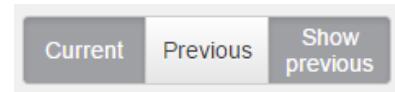
Trend-Protokolle, oder so genannte Datenprotokollierungen, ermöglichen dem Endbenutzer die Speicherung der ausgewählten Daten und das Vergleichen der verschiedenen Zeitabschnitte in der Vergangenheit.

Name	Object	Log type	Floating point pr...	1 minute data	Hourly data	Daily data	Log size	Created
Brightness office 1	0/0/0 (Brightness 1)	Absolute value	0	1 hour	5 years	30 days	344 KB	2014.02.06 14:48
Brightness office 2	0/0/1 (Brightness 2)	Absolute value	0	1 hour	30 days	30 days	7 KB	2014.02.06 14:48
Temperature	0/0/3 (Temperature)	Absolute value	2	1 hour	30 days	30 days	7 KB	2014.02.07 10:52
Humidity	0/0/2 (Humidity)	Absolute value	0	1 hour	30 days	30 days	7 KB	2014.02.07 10:53
CO2 level	0/0/1 (CO2)	Absolute value	0	1 hour	30 days	30 days	7 KB	2014.02.07 10:53

[+ Add new trend log](#)

Day Month Year

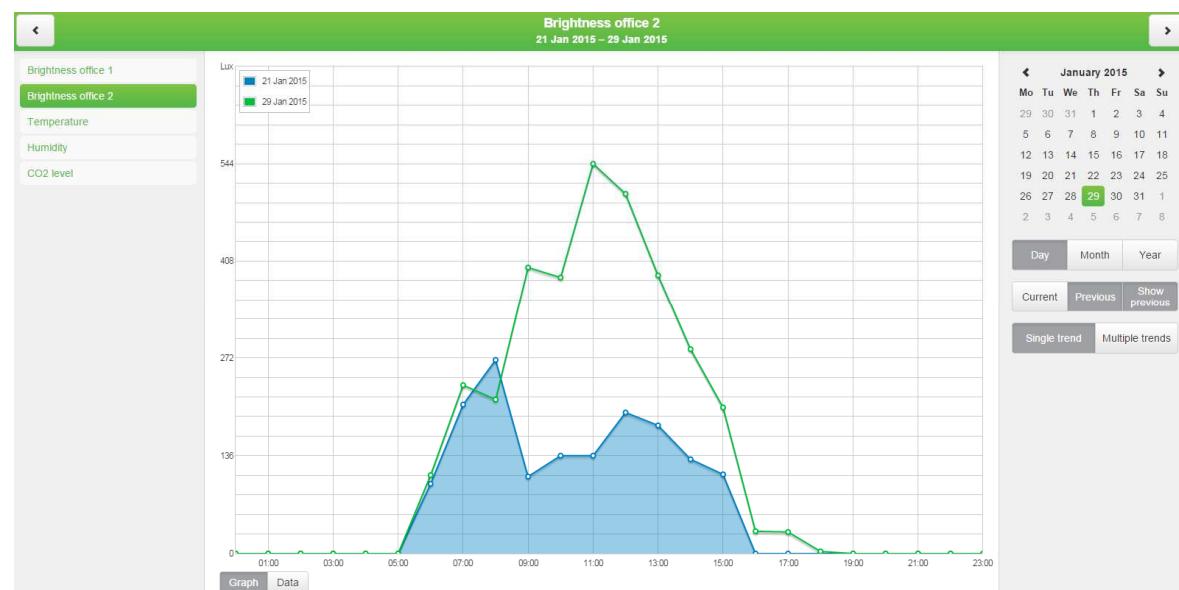
Trendzeitraum auswählen.

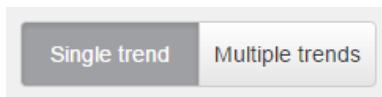


Vorherige anzeigen – Funktion früherer Werte für ausgewählten Zeitraum (Tag/Monat/Jahr) zum Datenvergleich aktivieren/deaktivieren.

Aktuell – Auswahl des aktuellen Datums.

Vorheriges – Auswahl des vorherigen Datums.

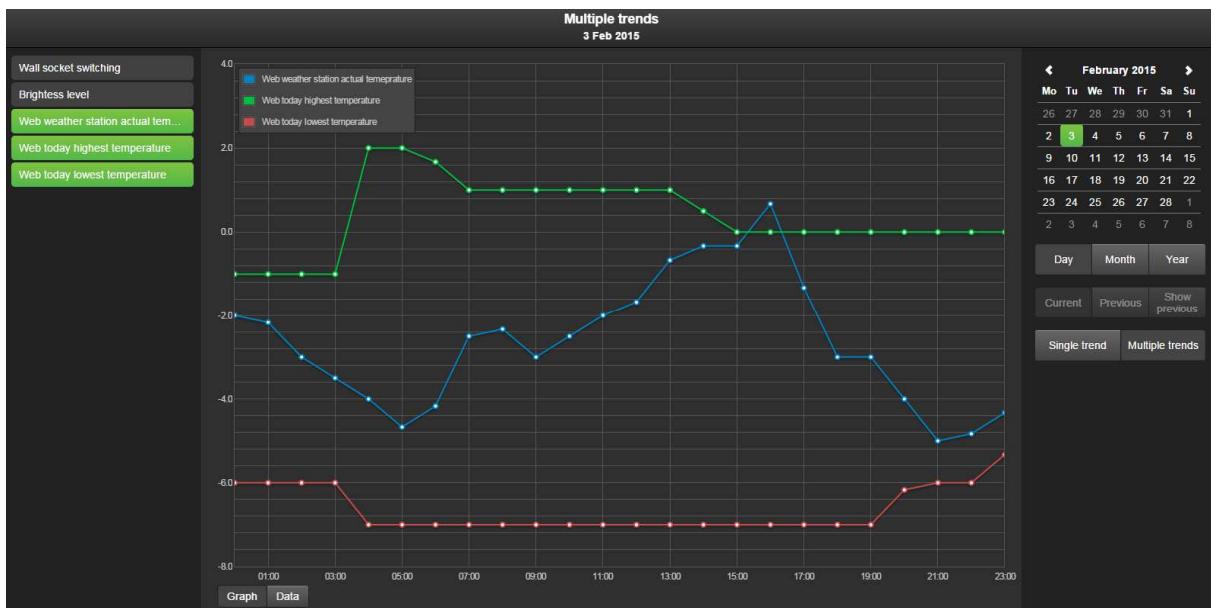




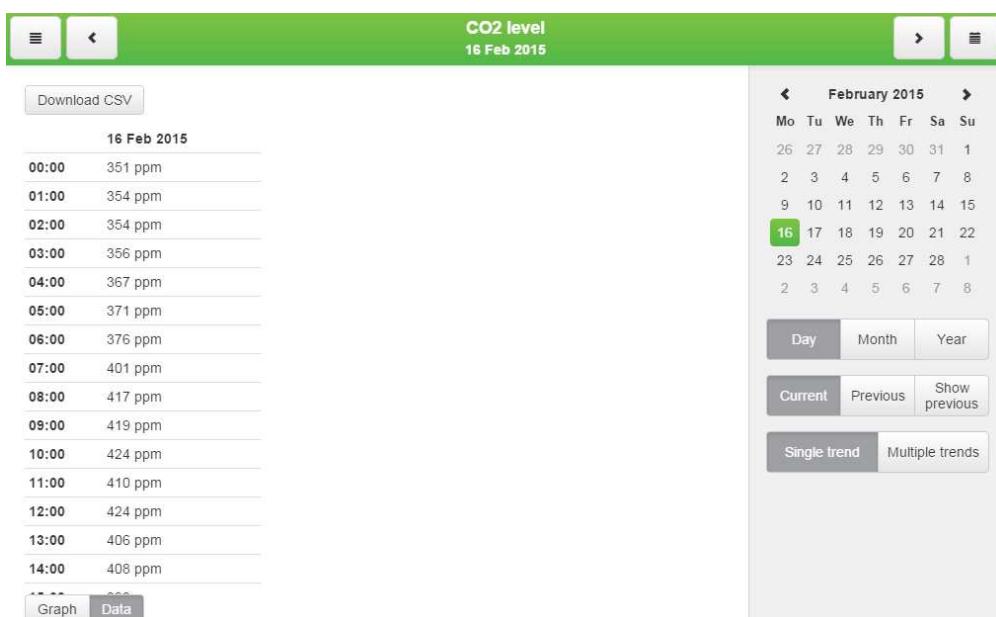
Auswahl zwischen der Anzeige eines einzelnen oder mehreren Trends (in Dunkles Design).



Die Auswahl der sichtbaren Trends wird in der Liste Trends durchgeführt.



Daten können außerdem im Zahlenformat angezeigt und zur späteren Verwendung im CSV-Format exportiert werden.



3.7.1 Neues Trendprotokoll hinzufügen.

Objekt – Wählen Sie aus der Liste der Objekte jenes aus, für das Sie Trends erstellen möchten.

Name – Name des Trends.

Protokolltyp – Typ des Protokolls.

Zähler - Wird zum Zählen der Daten verwendet.

Absoluter Wert – Speicher die aktuellen Auslesungen.

Fließkomma-Genauigkeit– Wenn das Objekt vom Typ Fließkomma ist, muss die Genauigkeit ausgewählt werden. Beispiel: 1.1111 = Genauigkeit ist 4.

1 Minute-Daten– Der Durchschnittswert von 1 Minute eines bestimmten Zeitintervalls, nach dem Daten im Trend angezeigt werden. Beispiel 1 Stunde – Der Schritt des Trends ist 1 Stunde, mit einem Durchschnitt von 60 Datenauslesungen.

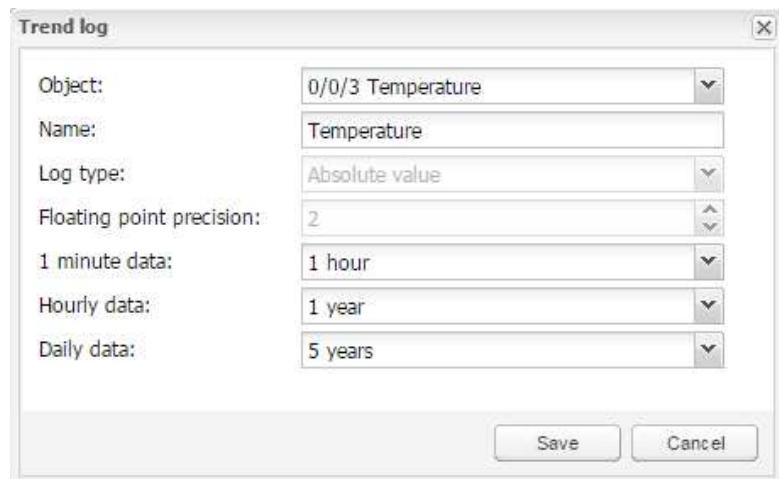
Stundendaten– Zeit der Speicherung von Stundendaten. (Max. 5 Jahre).

Tagesdaten– Zeit der Speicherung von Tagesdaten. (Max. 10 Jahre).

HINWEIS: Trendprotokolle werden im Speicher der internen SD-Karte gespeichert.

HINWEIS: Beachten Sie bitte, dass wenn der Protokolltyp auf **Zähler** gesetzt ist, dieser keine permanent abnehmende Tendenz haben kann. Verwenden Sie für diese Option einen **Zähler mit Delta negativ**.

Die **Trend**-Visualisierung wird in der **Smartphone Visualisierung** nicht unterstützt.



3.8 Vis. Struktur

Vis. Struktur wird zur Erstellung aller Gebäudeebenen und Visualisierungspläne verwendet. Zusätzlich können dort **Layouts** und **Widgets** für die Plan-Visualisierung erstellt werden.

Beim Start eines neuen Projekts sind nur Ordner für **Layouts** und **Widgets** sichtbar. Das Hinzufügen einer neuen Ebene ermöglicht dem Endbenutzer die Definition eines spezifischen **Plans** des Apartments. **Layouts** und **Widgets** sind zusätzliche Tools, die für die einfache Visualisierung nicht zwingend sind. Sie können in vielen anderen **Plänen** definiert und eingebunden werden.

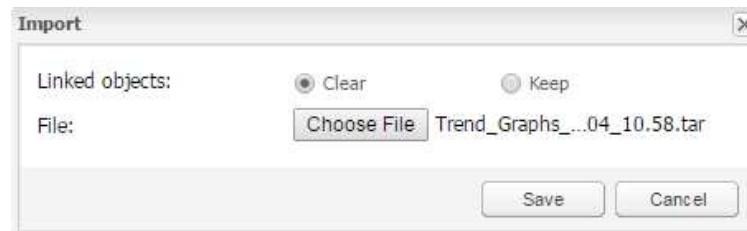
3.8.1 Ebenen

Zum Hinzufügen einer neuen Ebene drücken Sie auf die  Schaltfläche. Die Hauptebene ist normalerweise der Projektname. Zusätzliche Ebenen können später hinzugefügt werden.

Zum Importieren einer Ebene drücken Sie auf die  Schaltfläche.

Importierte Ebenen können hinzugefügt werden, oder den ursprünglichen Plan ersetzen.

Name	Visible	Description	Duplicate	Move up	Move d...	Add l...	Export
SE spaceLYnk office							
Overview	PC/Tablet, Smartphone						
Main Office	PC/Tablet, Smartphone						
Meeting Room	PC/Tablet						
Facility Management	PC/Tablet						
Garage	PC/Tablet, Smartphone						
Trend Graphs	PC/Tablet						
Schedulers	PC/Tablet						
Help	PC/Tablet						



3.8.2 Zweite Ebene

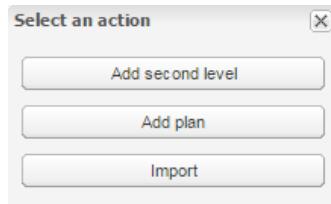
Die zweite Ebene wird in Gebäuden mit mehreren Etagen verwendet.

Drücken Sie auf die Schaltfläche  neben der Hauptebene, sollten Sie zusätzliche Ebenen benötigen.

Wählen **Sie Zweite Ebene** hinzufügen und geben Sie Namen und Sortierreihenfolge an.

Durch Drücken auf das Duplizieren-Symbol neben der Ebene dupliziert, oder zusammen mit Unterebenen und Plänen importiert werden.

Name	Visible	Description	Duplicate	Move up	Move d...	Add l...	Export
SE spaceLYnk office							
Overview	PC/Tablet, Smartphone						
Main Office	PC/Tablet, Smartphone						
Meeting Room	PC/Tablet						
Facility Management	PC/Tablet						
Garage	PC/Tablet, Smartphone						
Trend Graphs	PC/Tablet						
Schedulers	PC/Tablet						
Help	PC/Tablet						
2nd level							



3.8.3 Plan

Der Plan kann entweder einen Raum des Apartments mit kumulierten Funktionen oder nur einer Funktion (wie Beleuchtung oder Heizung) des kompletten Apartments anzeigen. Klicken Sie zum

Hinzufügen von Plänen auf die Schaltfläche neben der Ebene, unter der der Plan hinzugefügt werden soll und wählen **Sie Plan** hinzufügen.

Name – Name des Plans.

Plangröße – Größe des Plans. Vordefinierte Größen finden Sie im Menü .

Layout – Layout für diesen spezifischen Plan. Alle Objekte vom Layout, einschließlich Hintergrundfarbe und Planbild, werden in diesen bestimmten Plan dupliziert. Wenn diese nicht separat für diesen spezifischen Plan definiert sind. Vor dem Hinzufügen des Plans muss das Layout erstellt werden.

PC/Tablet-Visualisierung

[Anzeigen, Anzeigen und als Standard festlegen, Verbergen] – Sichtbarkeit für diesen bestimmten Plan in der PC/Tablet-Visualisierung.

Smartphone-Visualisierung

[Anzeigen, Anzeigen und als Standard festlegen, Verbergen] – Sichtbarkeit für diesen bestimmten Plan in der Smartphone-Visualisierung.

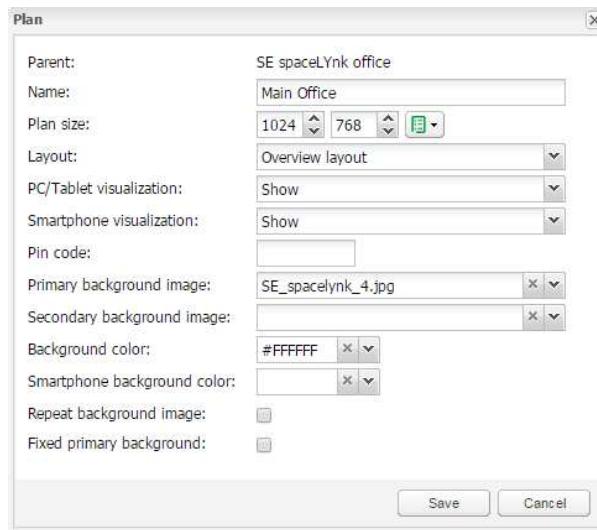
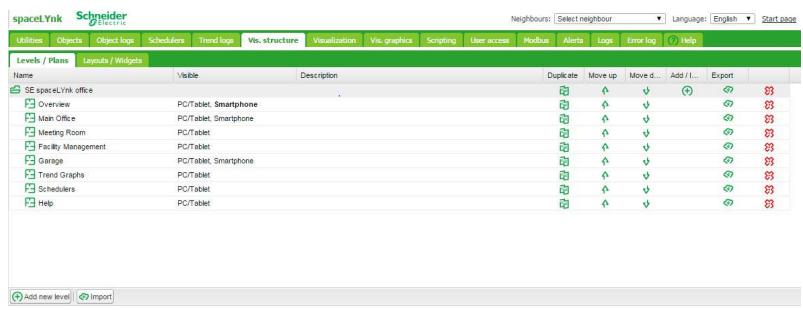
Pin code – Möglichkeit, jeden Plan mit einem PIN-Code zu sichern.

Primäres Hintergrundbild – Wählen Sie das primäre Hintergrundbild für diesen Plan.

Sekundäres Hintergrundbild – Wählen Sie den sekundären Hintergrund des Plans für eine Parallaxenoptik der Visualisierung. Wählen Sie den Hintergrund, den Sie vorher den **Vis.**

Grafiken hinzugefügt haben -> Bilder/Hintergründe.

Hintergrundfarbe – Wählen Sie eine Hintergrundfarbe für den Plan.



Hintergrundfarbe für Smartphone – Wählen Sie eine Hintergrundfarbe des Plans für die Smartphone-Visualisierung.

Hintergrundbild wiederholen – Entweder zur einmaligen Anzeige des Bildes, oder zur wiederholten Anzeige, um den kompletten Plan auszufüllen.

Fester, primärer Hintergrund – Statisches, primäres Hintergrundbild in Parallaxen-Projektion.

Durch Drücken auf das Duplizieren-Symbol neben dem Plan kann jeder Plan zusammen mit allen Komponenten in einem Plan dupliziert werden .

HINWEIS: Der Inhalt des erstellten Plans kann in der Registerkarte Visualisierung definiert werden.

Ein leerer Plan (ohne Objekte) wird in der Visualisierung nicht angezeigt.

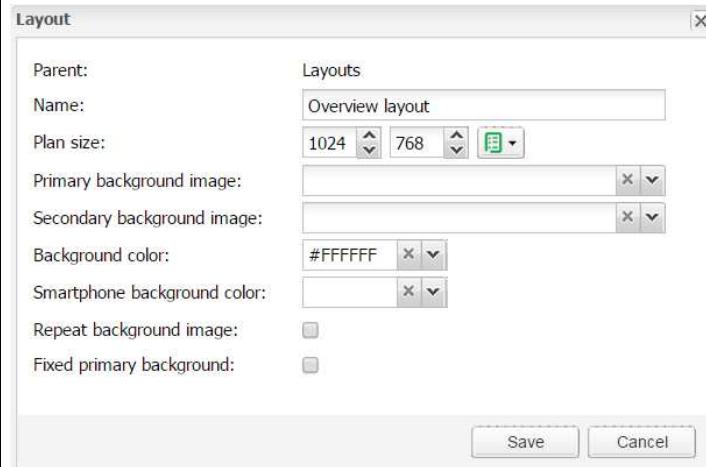
3.8.4 Layout

Layout ist der erweiterte Hintergrund für Pläne. Es kann jediges Objekt von Editor im Layout platziert werden, welches zu einem späteren Zeitpunkt einem oder mehreren Plänen hinzugefügt werden kann. Es werden alle Objekte aus dem Layout im Plan angezeigt. Allerdings sind alle Objekte im Plan über den Objekten aus dem Layout angeordnet.

Zum Hinzufügen eines Layout drücken Sie die Schaltfläche  neben dem Layout, oder  die Schaltfläche.

Jedes Layout kann zusammen mit allen Komponenten dupliziert werden. Drücken Sie dazu auf das Duplizieren-Symbol neben dem Layout .

HINWEIS: Der Inhalt dieses Layouts kann in der Registerkarte Visualisierung definiert werden.



3.8.5 Widget

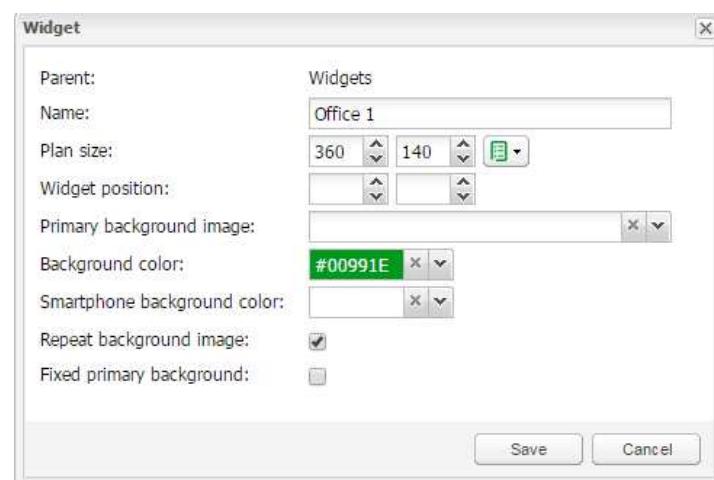
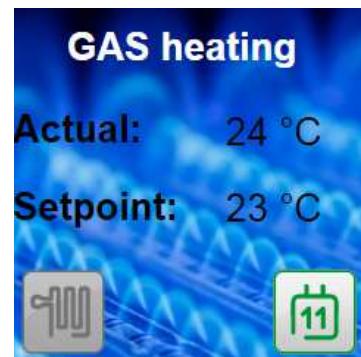
Ein Widget ist eine kleine Website, die einer Schaltfläche hinzugefügt werden kann, und bei Aktivierung erscheint.

Zum Hinzufügen von Widgets drücken Sie auf die Schaltfläche neben dem Ordner oder der Schaltfläche für Widgets. Jedes Widget kann zusammen mit allen Komponenten dupliziert werden. Drücken Sie dazu auf das Duplizieren-Symbol neben dem Widget .

HINWEIS: Der Inhalt dieses Widgets kann in der Registerkarte Visualisierung definiert werden.

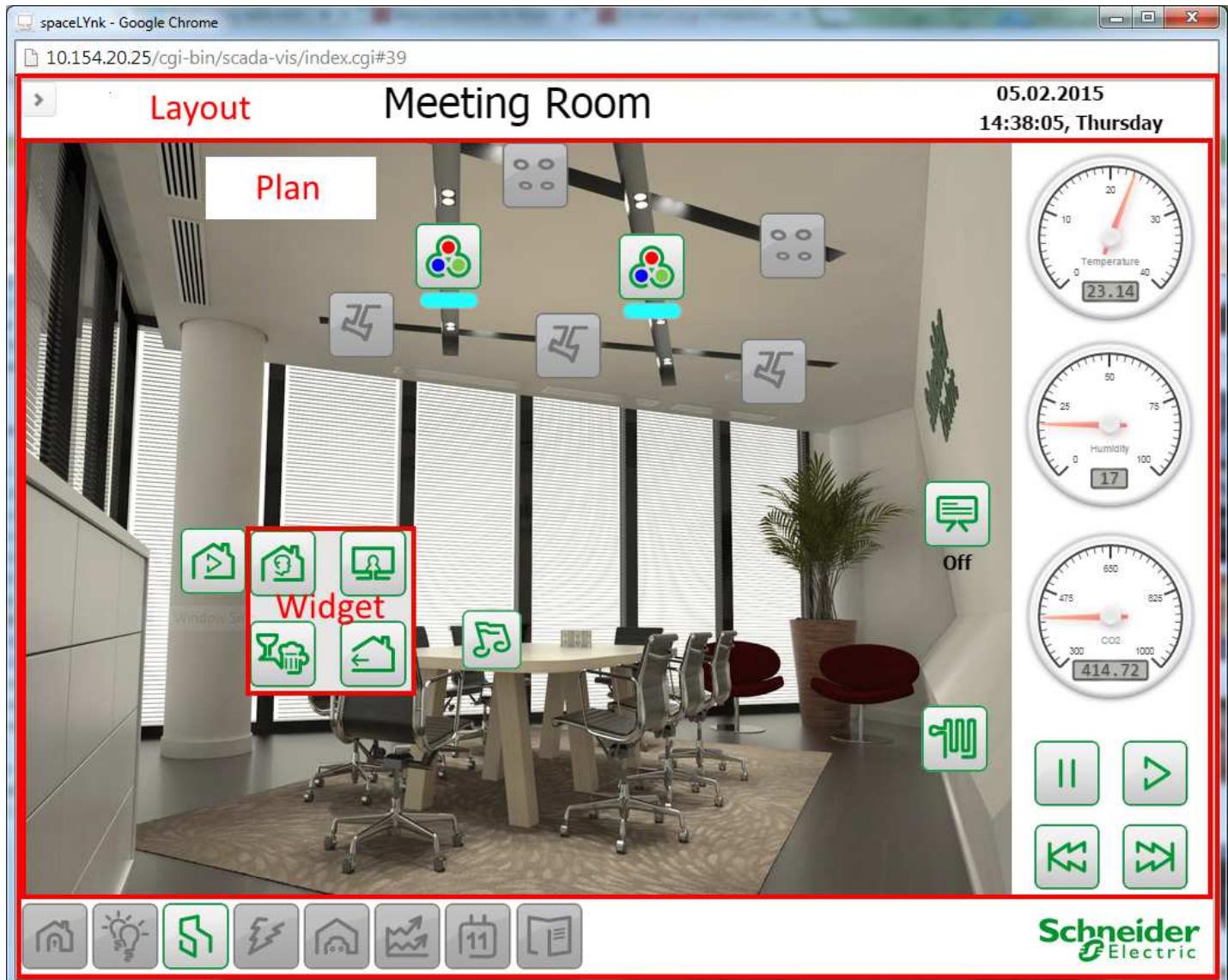
Die Größe des Widgets muss immer kleiner sein als der Plan, in dem das Widget platziert wird.

Ein leeres Widget (ohne Objekte) wird in der Visualisierung nicht angezeigt.

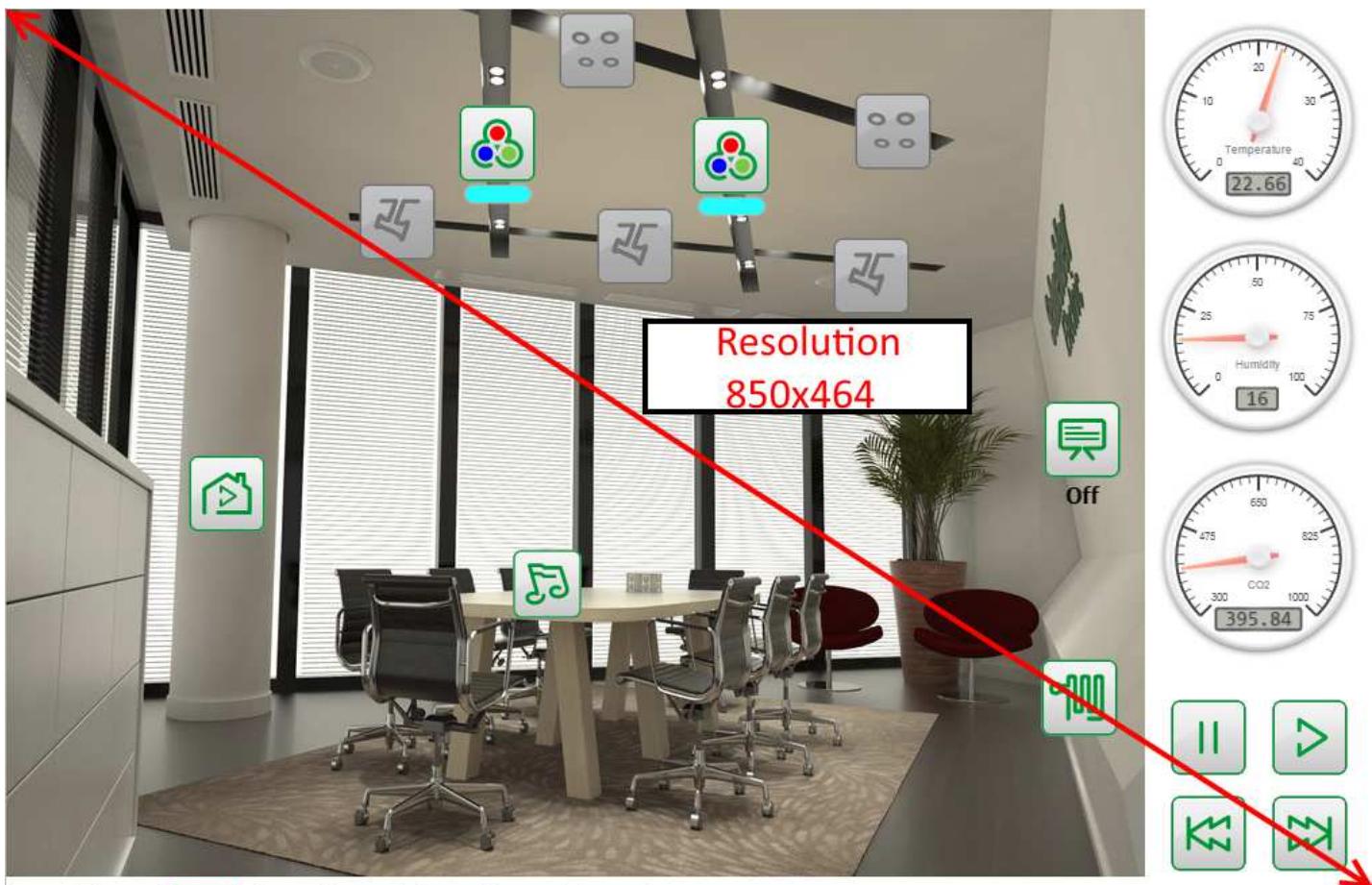


3.8.6 Beispiel einer Visualisierungsstruktur

3.8.7 Plan



3.8.8 Layout



3.8.9 Widget



3.8.10 Sortierung der Visualisierungsobjekte

Jedes Objekt in einer Visualisierung hat seine eigene Priorität. Diese ist von der Höchsten zur Niedrigsten Sortierung beschrieben:

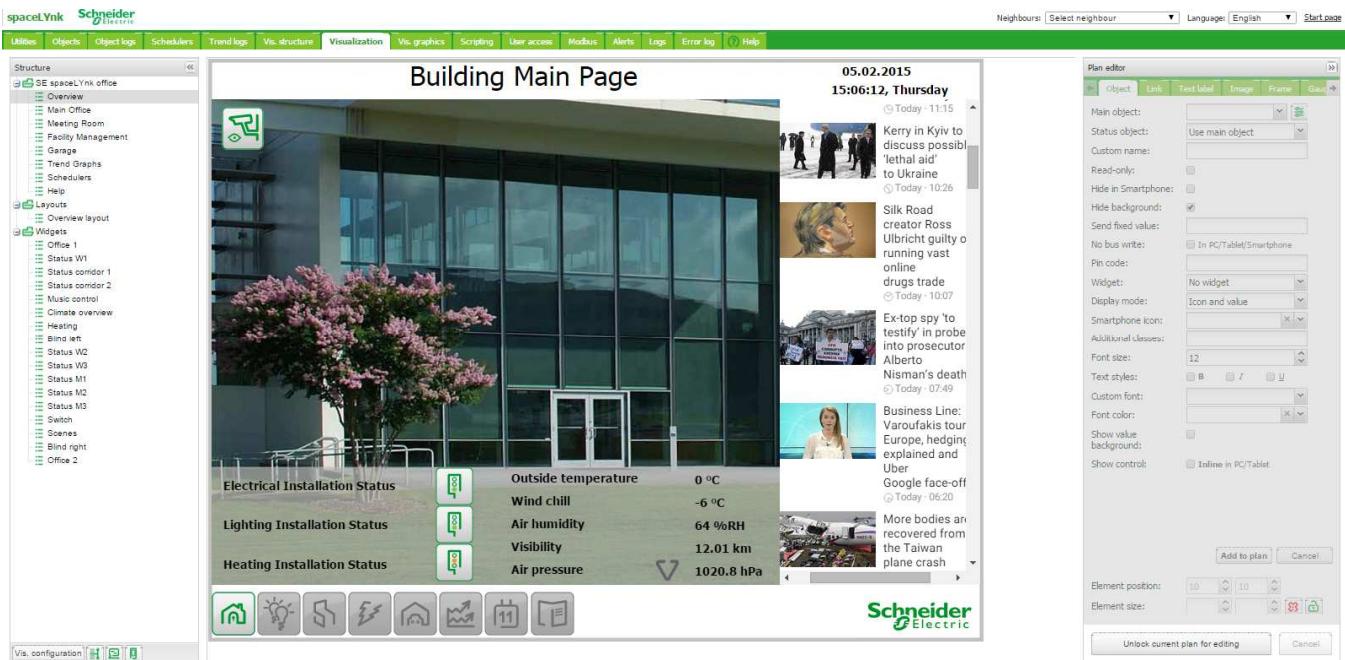
1. Objekt auf Plan
2. Plan-Verknüpfung als Text auf Plan
3. Plan-Verknüpfung als Symbol auf Plan
4. Kamera auf Plan
5. Diagramm auf Plan
6. Maß auf Plan
7. Beschriftung auf Plan
8. Bild auf Plan
9. Rahmen auf Plan
10. Objekt auf Layout
11. Plan-Verknüpfung als Text auf Layout
12. Plan-Verknüpfung als Symbol auf Layout
13. Kamera auf Layout
14. Diagramm auf Layout
15. Maß auf Layout
16. Beschriftung auf Layout
17. Bild auf Layout
18. Rahmen auf Layout
19. Hintergrund auf Plan
20. Hintergrund auf Layout

HINWEIS: Die Sortierung von Objekten mit der gleichen Priorität ist nicht definiert und kann in der Editor- und PC/Tablet-Visualisierung abweichen.

3.9 Visualisierung

Diese Registerkarte ist in drei Abschnitte aufgeteilt:

1. **Struktur** – Navigationsstruktur für Ebenen, Pläne, Widgets, die in der Registerkarte Visualisierungsstruktur erstellt wurden.
2. **Visualisierungskarte** – Aktuelles Visualisierungsfeld, in dem Sie alle Visualisierungskomponenten hinzufügen können.
3. **Plan Editor** – Alle Parameter der Komponente werden hier gesetzt.



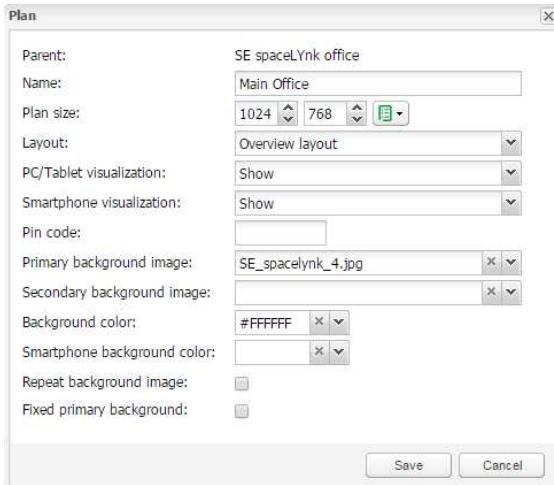
Durch Drücken auf das Symbol können beide Randleisten minimiert werden, wodurch der Plan, speziell bei kleinen Bildschirmen, besser zu betrachten ist.

3.9.1 Struktur

Verwenden Sie zur Navigation zwischen Plänen, Layouts und Widgets die Navigationsstruktur in der strukturierten Ansicht.

Folgende, zusätzliche Parameter sind im Bearbeitungsmodus verfügbar:

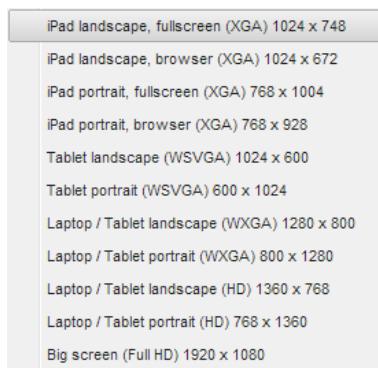
- Größe von Plan, Layouts und Widgets.
- Quellbild / Hintergrundfarbe



-

HINWEIS: Größe des Plans sollte korrekt vor dem Hintergrund positioniert werden. Die Größe des Widgets muss immer kleiner sein als der Plan, in dem das Widget platziert wird. Zum Ausrichten der Objekte immer die Komponentenposition verwenden.

Vordefinierte Größe des Plans:



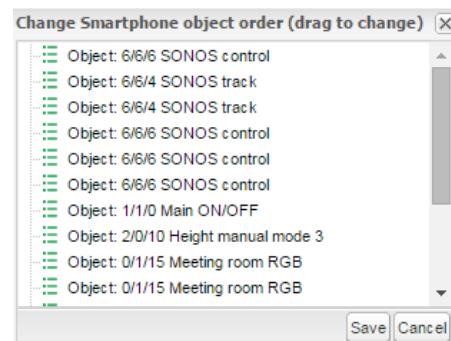
Zur Sortierung des Objekts in der Smartphone-Visualisierung drücken Sie auf die Schaltfläche Smartphone-Objekte neu

anordnen

Neben dem **Symbol für Smartphone-Objekte neu anordnen**

finden Sie zwei Symbole zum schnellen Zugriff auf:

- Vorschau in PC/Tablet.



- Vorschau in Smartphone.

3.9.2 Visualisierungskarte

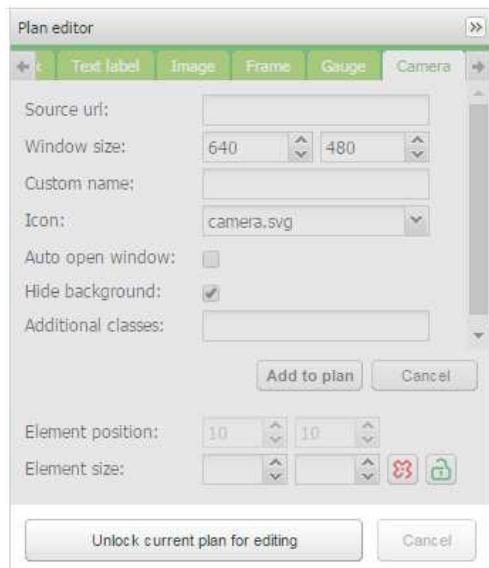
Jedes neu hinzugefügte Objekt erscheint an der oberen linken Ecke des Plans mit vorher im Objektmenü definiertem vertikalem und horizontalem Abstand.



Durch Ziehen an der Leiste an der oberen oder rechten Seite kann ein ausgewähltes Objekt geändert, gelöscht oder dupliziert werden (das duplizierte Objekt wird mit dem vordefinierten Abstand angezeigt).

3.9.3 Plan-Editor

Der **Plan-Editor** befindet sich auf der rechten Seite der Visualisierungskarte. Durch Drücken auf **Aktuellen Plan zur Bearbeitung entsperren** kann auf den Bearbeitungsmodus zugegriffen werden.



3.9.4 Objekt

In dieser Registerkarte wird jedes Kontroll- oder Überwachungsobjekt konfiguriert.

Unterschiedliche Datentypen haben unterschiedliche Parameter.

Durch Drücken auf das Symbol können die Parameter für die Objektvisualisierung geändert werden . Es bezieht sich auf **Vis.parameter** in der Registerkarte **Objekte**.

Hauptobjekt – Liste der existierenden Gruppenadressen im KNX/EIB-Bus, die in der Registerkarte **Objekte** zur Konfiguration verfügbar sind. Zur schnelleren Auswahl wird empfohlen, mit dem Schreiben von Gruppenadressen zu beginnen.

Statusobjekt – Liste der Statusobjekte im KNX/EIB-Bus. Als Status kann außerdem ein Steuerobjekt verwendet werden.

Benutzerdefinierter Name – Name für das Objekt. Benutzerdefinierter Name ist wichtig für die Smartphone-Visualisierung. Wenn dieser Name leer bleibt, wird stattdessen der Name der Gruppenadresse verwendet.

Schreibgeschützt – Das Objekt ist schreibgeschützt und kann nicht beschrieben (gesteuert) werden.

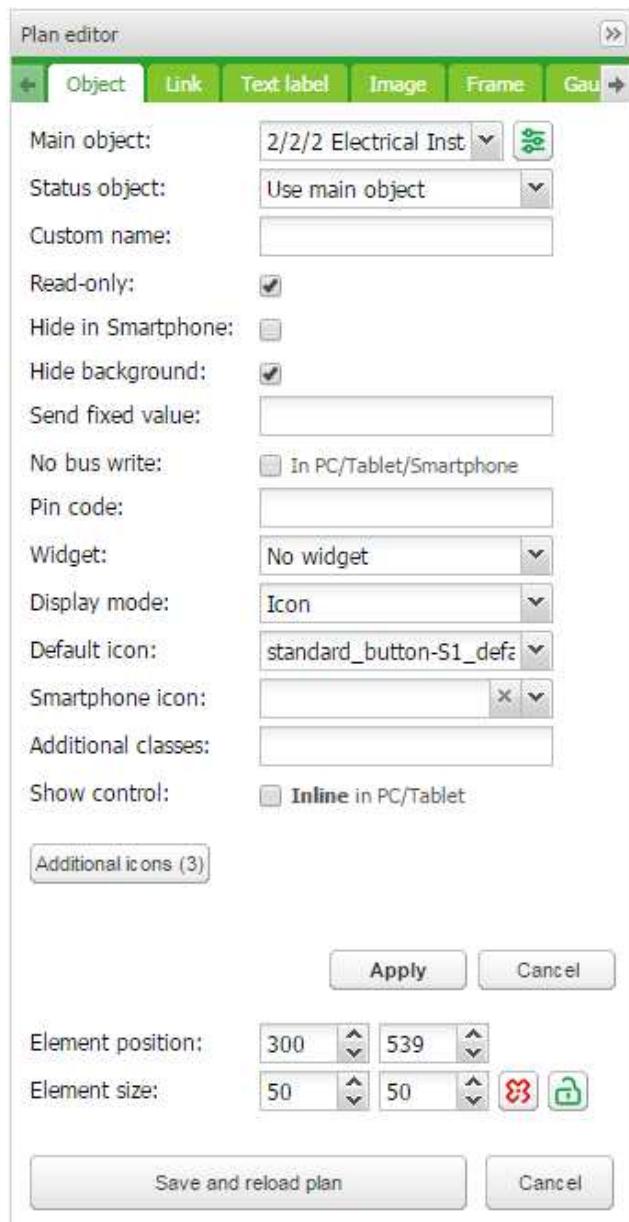
In Smartphone ausblenden – Dieses Objekt in der **Smartphone-Visualisierung** nicht anzeigen.

Hintergrund ausblenden – Hintergrund des Symbols ausblenden.

Festen Wert senden – Ermöglicht bei jedem Drücken auf das Objekt das Senden eines spezifischen Werts an den Bus.

Nicht in Bus schreiben – Wert wird nicht in den KNX-Bus geschrieben. Nützlich zur Auslösung eines Skripts mit Einschränkung der Bus-Last.

PIN-Code – Das Objekt kann durch Hinzufügen eines PIN-Codes geschützt werden. Bei jeder Änderung des Werts wird nach dem PIN-Code gefragt.



Widget– Ein Widget kann einer Schaltfläche hinzugefügt werden, die zuvor erst erstellt werden muss. Das Widget kann nur in der PC/Tablet-Visualisierung, aber nicht im Bearbeitungsmodus getestet werden.

Anzeigemodus [Symbol und Wert; Symbol; Wert] – Wie das Objekt angezeigt wird.

Smartphone Symbol– Standardsymbol für Smartphone, wenn sich dieses vom Symbol für PC/Tablet unterscheidet.

An-Symbol– Symbol zur Anzeige des Status An für Objekte des Typs Binär.

Aus-Symbol– Symbol zur Anzeige des Status Aus für Objekte des Typs Binär.

Zusätzliche Klasse– Erstellen Sie zusätzliche Klassen, die in benutzerdefinierten CSS-Dateien verwendet werden können, um eine bestimmte Gruppe grafischer Objekte zu modifizieren.

Schriftgröße– Die Textart für die Wertanzeige kann definiert werden.

Textarten– [Fett / Kursiv / Unterstreichen]– Option.

Benutzerdefinierte Schriftart– Auswahl aus installierten Schriften.

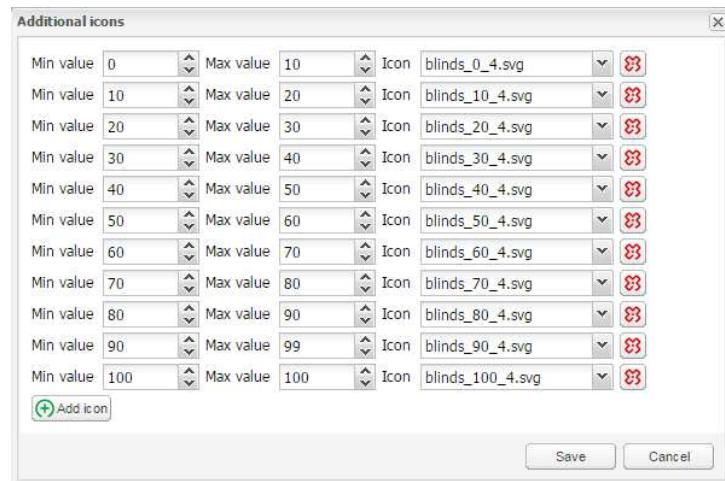
Schriftgröße– Größe der ausgewählten Schrift.

Steuerung anzeigen– Wenn ausgewählt ändert sich jede Grafik einer Schaltfläche für die Steuerung von einem Symbol in einen Schalter . Nur in der PC/Tablet-Visualisierung sichtbar.

Bei der Spezifikation der Parameter erscheint eine zusätzliche Schaltfläche für Objekte des Typs Wert – **Zusätzliche Symbole**. In diesem Fenster können verschiedene Symbole für verschiedene Objektwerte definiert werden.

Die Textart für die Wertanzeige kann definiert werden.

Nach der Definition der Objektparameter drücken Sie auf die Schaltfläche **Zu Plan hinzufügen**, und das neu erstellte Objekt erscheint. Das Objekt kann zu jedem beliebigen Ort im Plan bewegt



werden.

HINWEIS: Das Objekt funktioniert im Bearbeitungsmodus nicht. Nach dem Hinzufügen aller nötigen Objekte drücken **Sie auf die Schaltfläche Speichern und Etagenplan erneut laden**, damit die Objekte zu funktionieren beginnen.

Im Bearbeitungsmodus kann jedes hinzugefügte Objekt durch Klicken darauf bearbeitet werden.

Drücken  Sie nach jeder Änderung auf die Schaltfläche .

Durch Drücken auf die Schaltfläche an der linken

 Seite kann jedes Objekt dupliziert werden.

 Drücken auf die Schaltfläche ändert die Objektparameter auf Standardeinstellung.

Elementposition–Kann manuell oder durch Drag and Drop des Objekts für X- und Y-Position im Plan hinzugefügt werden.

Elementgröße– Kann manuell, oder durch Ziehen der vertikalen oder horizontalen Leiste des Objekts hinzugefügt werden.

Durch Drücken auf das Zurücksetzen-Symbol kann die Standardgröße wiederhergestellt werden -



Durch Drücken auf das Schloss-Symbol kann das Seitenverhältnis eines Objekts gesperrt werden



3.9.5 Verknüpfung

Zur Verbesserung der Bedienfreundlichkeit der Visualisierung sind Plan-Verknüpfungen integriert. Der Karte können spezielle Symbole hinzugefügt werden, die als Verknüpfung zu anderen Plänen dienen.

Verknüpfen mit – Verknüpfung zum Plan auswählen.

Benutzerdefinierter Name –Name für die Verknüpfung.

In Smartphone ausblenden – Diese Plan-Verknüpfung in der Smartphone-Visualisierung nicht anzeigen.

Hintergrund ausblenden – Hintergrund des Symbols ausblenden.

Anzeigemodus [Symbol; Wert] – Wie die Plan-Verknüpfung angezeigt wird.

Symbol – Das in der Visualisierung angezeigte Symbol. Wenn nur Text ausgewählt ist, werden Textparameter ausgewählt.

Aktiver Status-Symbol – Wenn nicht ausgewählt ist das Symbol für den aktiven Plan verfügbar.

Zusätzliche Klasse – Erstellen Sie zusätzliche Klassen, die in benutzerdefinierten CSS-Dateien verwendet werden können, um eine bestimmte Gruppe grafischer Objekte zu modifizieren.

Schriftgröße – Größe der Schrift.

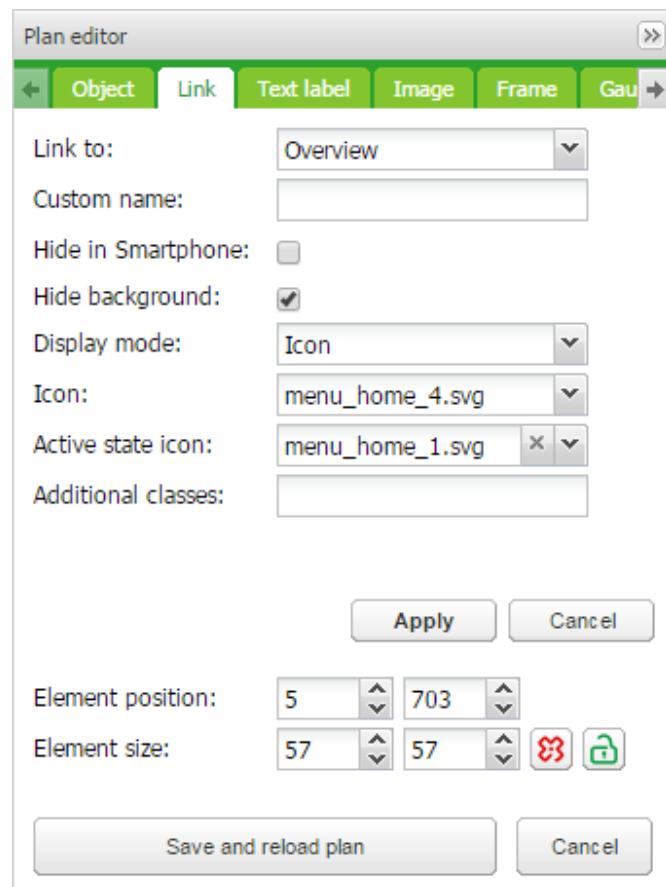
Textart – Textart – Fett, Kursiv, Unterstrichen.

Benutzerdefinierte Schriftart – Name der Schrift.

Schriftfarbe – Schriftfarbe.

Größe und Position des Elements – Siehe **Plan editor > Registerkarte Objekt**.

HINWEIS: Zur Erstellung einer Verknüpfung für Menü und Plan wird die Verwendung von Layout empfohlen. Beim Hinzufügen zu unterschiedlichen Plänen und später bei der Durchführung von Änderungen kann Zeit eingespart werden. Dies würde beim Hinzufügen unterschiedlicher Pläne Zeit sparen und ist nützlich, wenn Änderungen nötig sind.



3.9.6 Kamera

spaceLYnk unterstützt die Integration einer IP-Web-Kamera von Dritten in seine Visualisierung.

HINWEIS: Nur Kameras, die HTTP MJPEG-Streaming im Webbrower unterstützen, können visualisiert werden.

Quell-URL – Quelladresse des Video-Streams.

Fenstergröße – Breite und Höhe zur Anzeige des Bildes.

Benutzerdefinierter Name – Name für das Objekt.

Fenster automatisch öffnen – Fenster für das Video automatisch öffnen, wenn der Plan geöffnet ist.

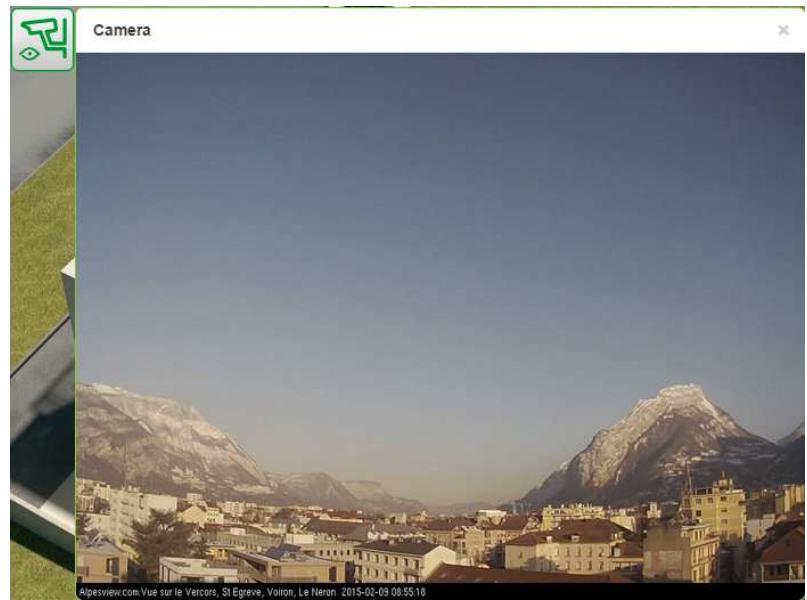
Zusätzliche Klasse – Erstellen Sie zusätzliche Klassen, die in benutzerdefinierten CSS-Dateien verwendet werden können, um eine bestimmte Gruppe grafischer Objekte zu modifizieren.

Hintergrund ausblenden – Hintergrund des Symbols ausblenden.

Sortierreihenfolge – Sortierung der Kamera für die Touch-Visualisierung.

HINWEIS:

- Sollte die IP-Kamera Benutzername und Kennwort benötigen, so geben Sie die entsprechende URL ein:
<http://USER:PASSWORD@IP>
- Sie können das Symbol oder die Beschriftung der Kamera nach Ihren Wünschen ändern.
- spaceLYnk arbeitet nur als Umleitungsstream von Kamera zu Browser. Sollte der Stream nicht funktionieren, so ist das ein Problem mit dem Webbrower, nicht mit spaceLYnk.
- Bei einem Problem mit der Kamera prüfen, ob der Videostream im Browser verfügbar ist.
- Wenn die Kamera extern verfügbar ist, muss der Port für das IP der Kamera durch den Router weitergeleitet werden. Beim Hinzufügen einer externen Kamera muss das IP mit dem korrekten Port verwendet werden ([IP:Port](#)). Bei Verwendung eines lokalen IP ist die Kamera extern nicht verfügbar.
- Wenn der Direkt-Video-Stream vom Hersteller der Kamera verdeckt ist, diesen bitte kontaktieren.



3.9.7 Diagramme

Zur Überwachung des aktuellen und alten Werts von Objekten des Typs Skalierung können Echtzeit-Diagramme in das Visualisierungssystem eingebunden werden. Vergewissern Sie sich, dass in der Registerkarte **Objekt** die Protokollierung für Objekte aktiviert ist, für die Werte im Diagramm angezeigt werden sollen.

Datenobjekt – Gruppenadresse des Objekts.

Damit das **Diagramm** in der **Visualisierung** aktiv ist, muss die Option **Protokollierung** für das Objekt aktiviert sein.

Benutzerdefinierter Name – Name des Objekts.

Symbol – Symbol, um das Diagramm zu starten.

Fenstergröße – Breite und Höhe zur Anzeige des Bildes.

Anzahl der Punkte – Anzahl der im Diagramm angezeigten Datenpunkte.

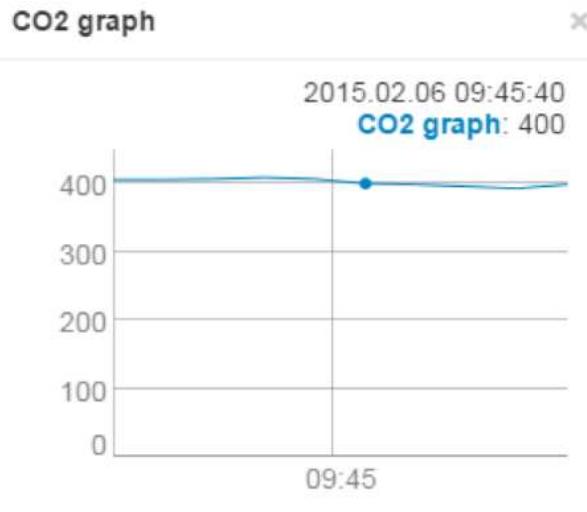
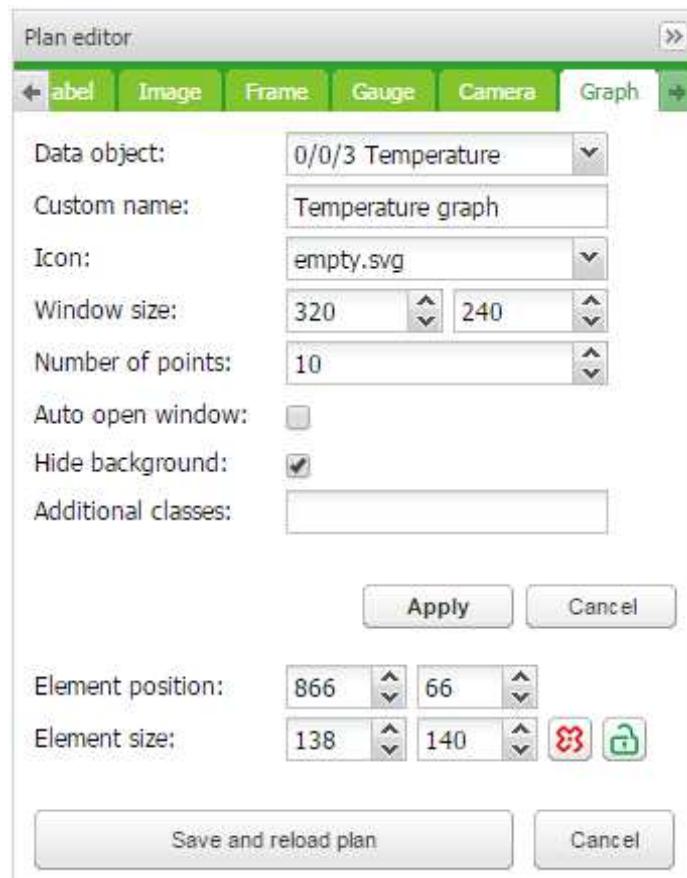
Fenster automatisch öffnen – Das Fenster mit dem Diagramm wird automatisch geöffnet.

Hintergrund ausblenden – Hintergrund des Symbols ausblenden.

Zusätzliche Klasse – Erstellen Sie zusätzliche Klassen, die in benutzerdefinierten CSS-Dateien verwendet werden können, um eine bestimmte Gruppe grafischer Objekte zu modifizieren.

Nach der Definition der Parameter für das Diagramm drücken Sie auf die Schaltfläche **Zu Plan hinzufügen**, und das neu erstellte Objekt erscheint. Das Objekt kann zu einem gewünschten Ort verschoben werden.

HINWEIS: Das Diagramm funktioniert im Bearbeitungsmodus nicht. **Drücken Sie auf die Schaltfläche Speichern und Plan erneut laden**, damit die Objekte zu funktionieren beginnen. (Mit Verzögerung zum Erhalt relevanter Daten.) Objekte, von denen die Daten erhalten werden, müssen in den Eigenschaften des **Objekts als protokolliert** festgelegt sein.



3.9.8 Beschriftung

Beschriftungen können hinzugefügt und in der Visualisierungskarte verschoben werden.

Text – Beschriftungstext

Schriftgröße – Schriftgröße des Beschriftungstextes

Textart – Art des Textes – Fett, Kursiv, Unterstrichen.

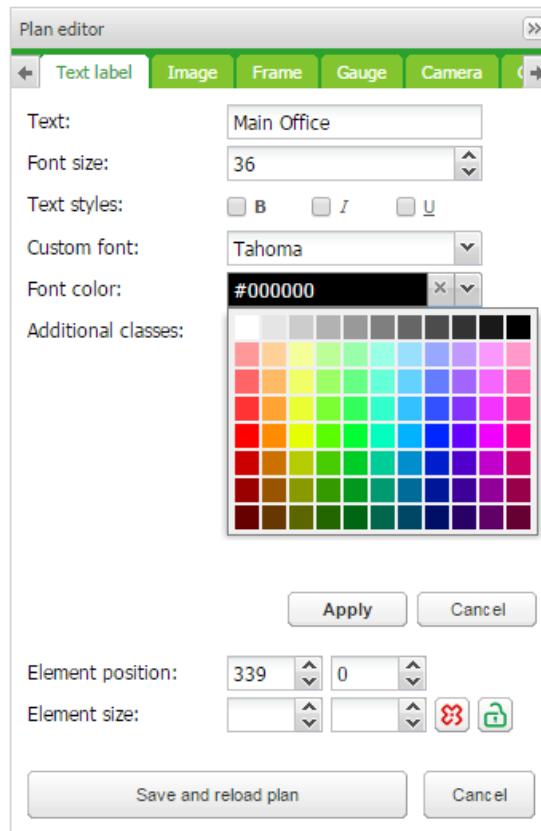
Benutzerdefinierte Schriftart – Name der Schrift.

Schriftfarbe – Schriftfarbe des Beschriftungstextes.

Zusätzliche Klasse – Für benutzerdefinierte CSS-Stile.

Nach der Definition der Parameter für die Beschriftung drücken Sie auf die **Schaltfläche Neues Objekt** hinzufügen, und die neu erstellte Beschriftung erscheint in der Karte. Das Objekt kann zu einem gewünschten Ort verschoben werden. **Drücken Sie auf die Schaltfläche Speichern und Etagenplan erneut laden**, damit die Beschriftungen zu funktionieren beginnen

Die letzten beiden Zeilen der Farbpalette sind die vordefinierten Unternehmensfarben von Schneider Electric.



3.9.9 Bild

Bildabschnitt erlaubt Ihnen, der Visualisierungskarte Bilder aus dem lokalen Speicher oder aus dem Internet hinzuzufügen. Externe Bilder sind beispielsweise nützlich, um dynamische Bilder der Wettervorhersage einzubinden.

Bildquelle [Lokal, Remote] – Wählen Sie die Bildquelle aus.

Bild auswählen – Wählen Sie ein Bild, welches Sie vorher den **Vis. Grafiken hinzugefügt haben -> Bilder/Hintergründe**.

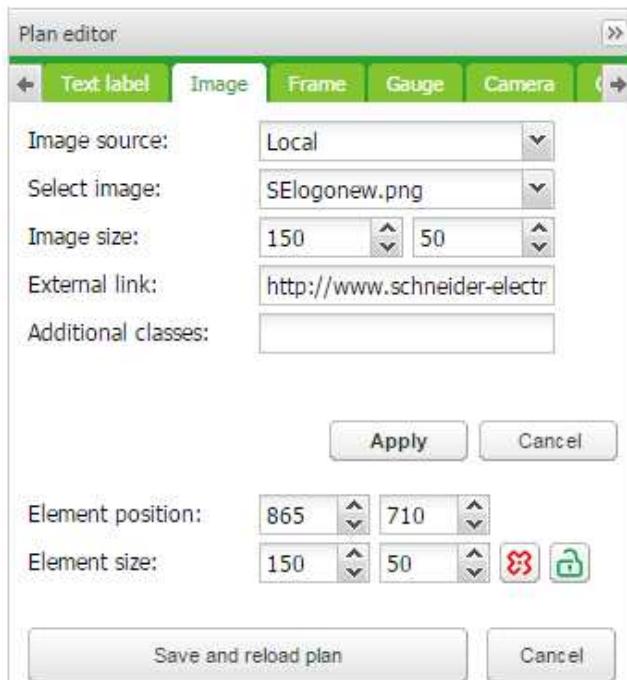
Bild-URL – Quell-URL des Bildes.

Breite – Breite des Bildes.

Höhe – Höhe des Bildes.

Externe Verknüpfung – Externe Verknüpfungs-URL beim Drücken auf das Bildbeispiel:
<http://www.schneider-electric.com>

Zusätzliche Klasse – Für benutzerdefinierte CSS-Stile.



Nach der Definition der Parameter für das Bild drücken **Sie auf die Schaltfläche Zu Plan hinzufügen**, und das neu erstellte Objekt erscheint in der Karte. Das Objekt kann zu einem gewünschten Ort verschoben werden. Durch halten und bewegen des Bildrandes kann das Bild nach Belieben geändert werden. Zur Anwendung der Änderungen drücken Sie auf **Speichern und Plan erneut laden**.

3.9.10 Messgerät

Messgerät erlaubt die dynamische Art der Visualisierung und das Verändern des Objektwerts im Messgerät.

Datenobjekt – KNX-Gruppenadresse.

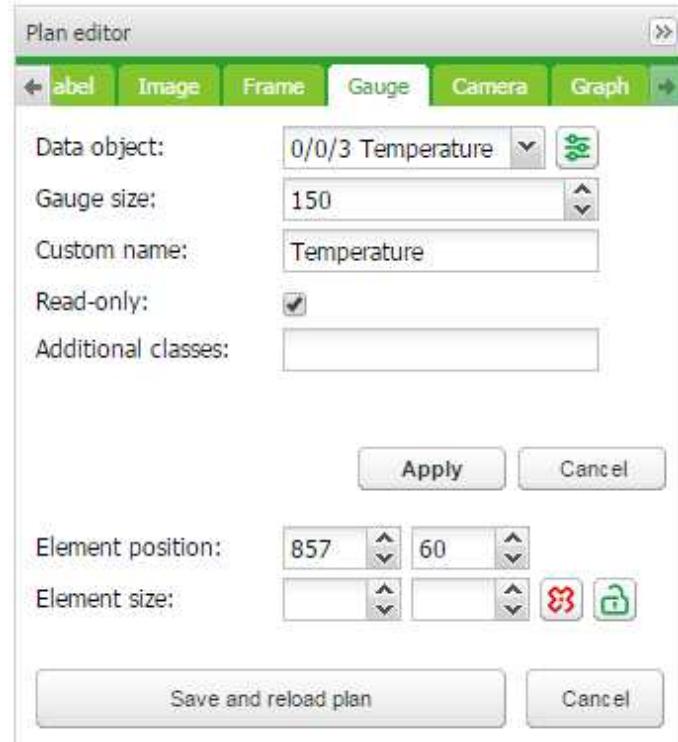
Messgerätgröße – Größe des Messgeräts.

Benutzerdefinierter Name – Benutzerdefinierter Name für das Objekt.

Schreibgeschützt – Messgerät mit Schreibschutz versehen.

Zusätzliche Klasse – Erstellen Sie zusätzliche Klassen, die in benutzerdefinierten CSS-Dateien verwendet werden können, um eine bestimmte Gruppe grafischer Objekte zu modifizieren.

Nach der Definition der Parameter für das Messgerät drücken **Sie auf die Schaltfläche Zu Plan hinzufügen**, und das neu erstellte Objekt erscheint in der Karte. Das Objekt kann zu einem gewünschten Ort verschoben werden. Drücken **Sie auf die Schaltfläche Speichern und Plan erneut laden**, damit das Messgerät zu funktionieren beginnt.



3.9.11 Rahmen

Mit Rahmen können externe und interne Websites in der Visualisierung angezeigt werden. **Planer** und **Trends** können in den Rahmen integriert werden.

Quelle – Wählen Sie Planer, Trendprotokoll, oder externe URL.

Url: - Quell-URL einer externen Website.

Rahmengröße: Breite/Höhe des Rahmens

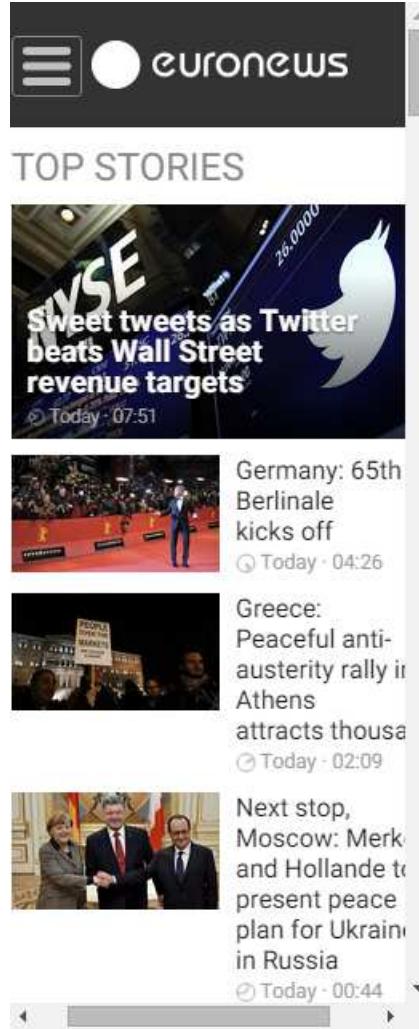
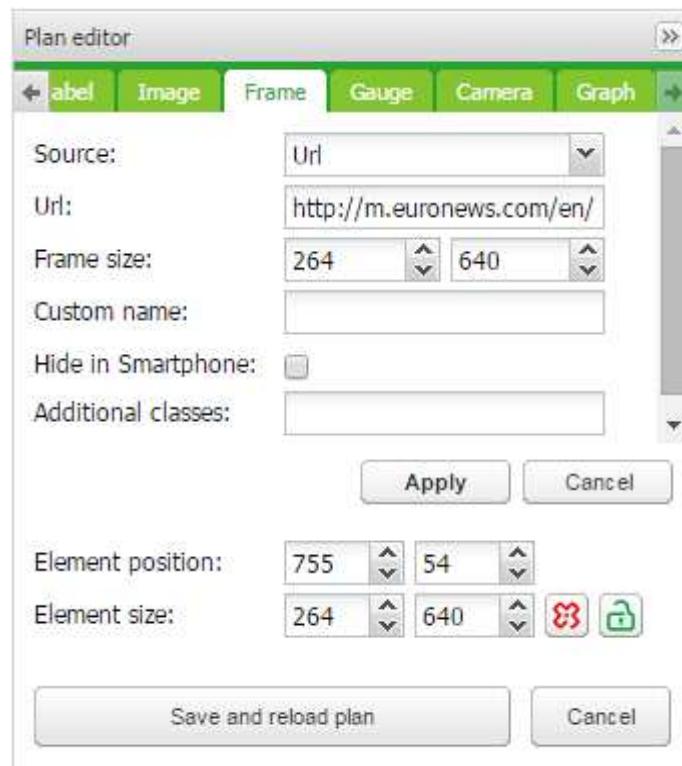
Benutzerdefinierter Name - Spezifizieren Sie den Titel für den Rahmen.

In Smartphone ausblenden - Wenn markiert, ist der Rahmen in der Smartphone-Visualisierung nicht angezeigt.

Nach der Definition der Parameter für den Rahmen drücken **Sie auf die Schaltfläche Zu Plan hinzufügen**, und das neu erstellte Objekt erscheint in der Karte. Der Rahmen kann zu einem gewünschten Ort verschoben werden. Durch Halten und Bewegen des Rahmens kann dieser nach Belieben geändert werden. Drücken **Sie auf die Schaltfläche Speichern und Plan erneut laden**, damit der Rahmen zu funktionieren beginnt.

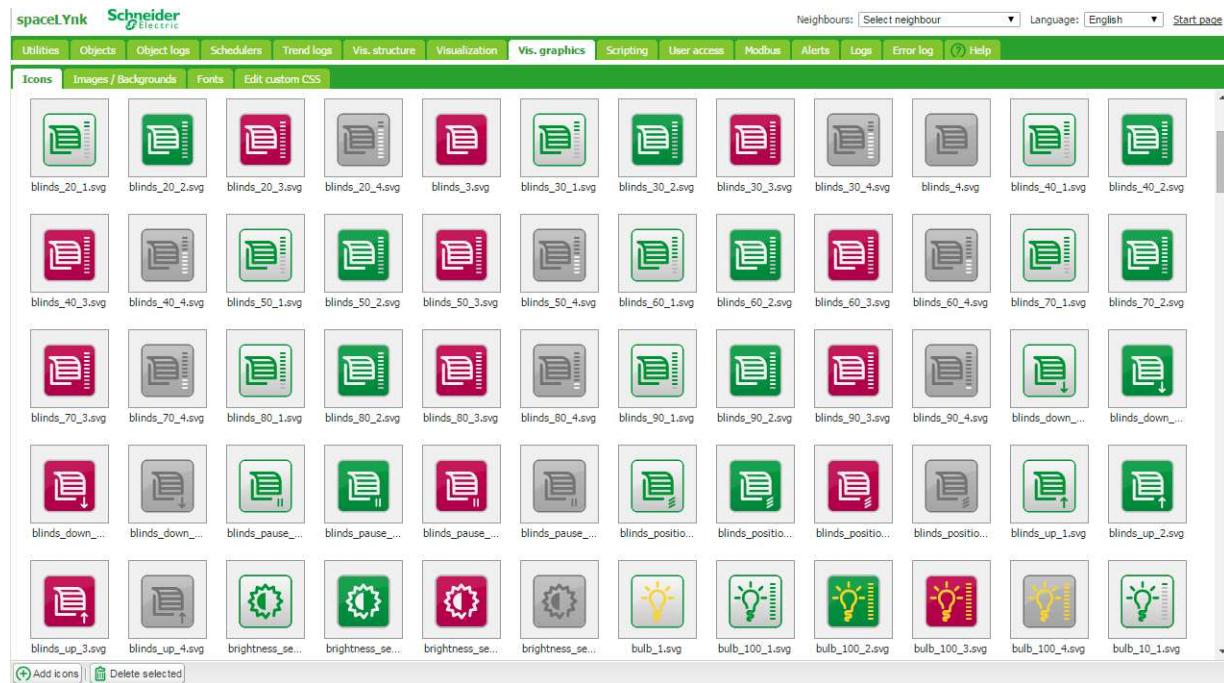
HINWEIS:

- Einige Websites verwenden Java-Script, was eine Verwendung des Rahmens verhindert. Sollte dies implementiert sein, so öffnet sich die Website über den kompletten Bildschirm, anstatt im Rahmen.
- Bei der Verwendung von Planer oder Trend wird empfohlen, den Rahmen auf die maximale Breite zu setzen. Minimal empfohlene Breite ist 1024.
- Rahmen ist nur in der PC/Tablet-Visualisierung sichtbar.
- Erlauben Sie die Anzeige von Planer oder Trend in der Smartphone-Visualisierung nicht. Einstellungen sind in **Vis. structure** des betreffenden Plans verfügbar.



3.10 Vis. graphics

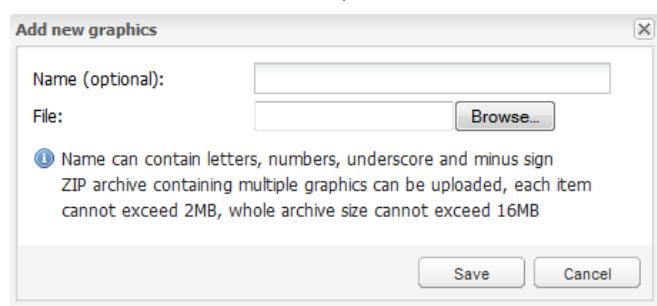
Diese **Registerkarte** ist in drei Abschnitte aufgeteilt. Symbole: Hier können alle Objektsymbole gefunden werden. **Bilder/Hintergründe**: Hier können alle lokal gespeicherten Bilder gefunden werden. **Benutzerdefiniertes CSS bearbeiten**: Hier können benutzerdefinierte CSS-Formatvorlagen erstellt oder bearbeitet werden.



Zum Hinzufügen eines neuen Eintrags drücken Sie auf die Schaltfläche **Neues Symbol hinzufügen**. Das System akzeptiert Symbole jeder Größe.

Die Formate JPEG, GIF, PNG und SVG werden unterstützt. Der Name kann Buchstaben, Nummern, Unterstriche und Minus-Zeichen beinhalten.

Es können ZIP-Archive mit mehreren Grafiken hochgeladen werden, wobei kein Element 2 MB überschreiten darf. Die komplette Größe des Archivs darf 16 MB nicht überschreiten.



Name (optional) – Der Name des Symbols. Beim Hinzufügen eines neuen Objekts erscheint dies in der Liste. Es kann Buchstaben, Nummern, Unterstriche und Minus-Zeichen beinhalten.

Datei – Speicherort der Symboldatei.

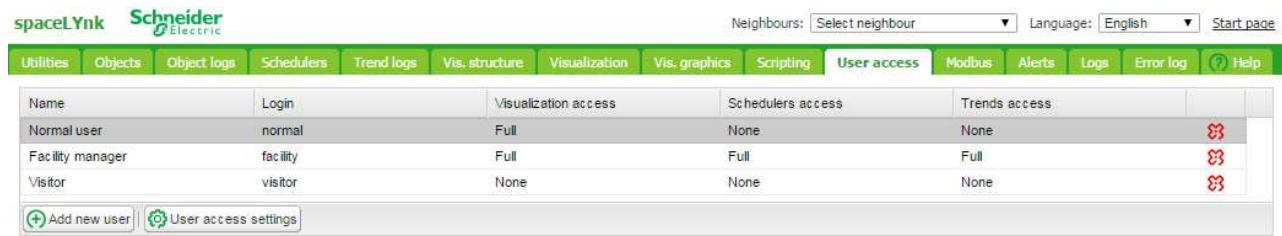
Eine CSSS-Formatvorlage kann durch Hochladen einer neuen Datei geändert werden. CSS definieren alle Schaltflächen, die Smartphone-Visualisierung, Planer und Trends. Für weitere Informationen

über die Modifikation von CSS-Dateien kontaktieren Sie bitte Ihr örtliches Front Office, um zusätzliche Dokumente zu erhalten.

HINWEIS:Nach dem Hochladen einer neuen CSS-Datei, bitte den Browser-Cache leeren.

3.11 Benutzerzugriff

Das Menü Benutzerzugriff ermöglicht die Erstellung und Verwaltung von Benutzerkonten.



The screenshot shows a web-based interface for managing user access. At the top, there are navigation links: Utilities, Objects, Object logs, Schedulers, Trend logs, Vis. structure, Visualization, Vis. graphics, Scripting, User access (which is highlighted in green), Modbus, Alerts, Logs, Error log, and Help. Above the main content area, there are dropdown menus for Neighbours (Select neighbour) and Language (English). A "Start page" link is also present. The main content is a table titled "User access" with the following columns: Name, Login, Visualization access, Schedulers access, Trends access, and three red crossed-out icons. There are four rows of data:

Name	Login	Visualization access	Schedulers access	Trends access	
Normal user	normal	Full	None	None	
Facility manager	facility	Full	Full	Full	
Visitor	visitor	None	None	None	

At the bottom left of the table are two buttons: "Add new user" and "User access settings".

Neuen Benutzer hinzufügen

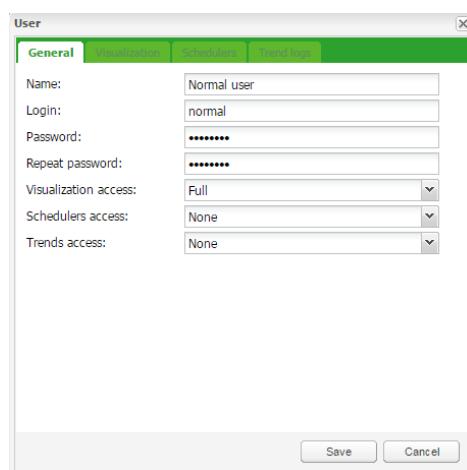
Zum Hinzufügen eines neuen Benutzers klicken Sie auf das Symbol.

Benutzername – Name des Kontos.

Benutzername – Länge von 2 bis zu 20 Zeichen.

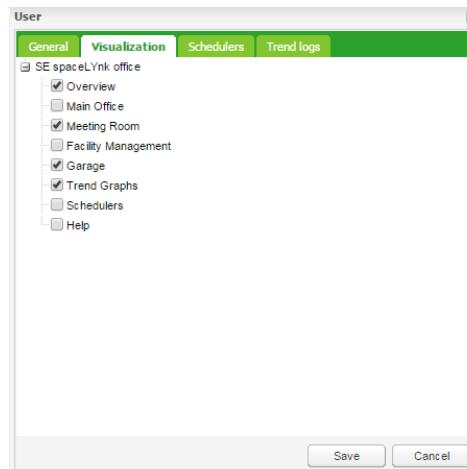
Erlaubte Zeichen: „-“, „_“, „a-z“, „A-Z“, „0-9“.

Kennwort – Länge von 6 bis zu 20 Zeichen. Alle Zeichen sind erlaubt.



Zugriff auf Visualisierung/Planer/Trends –

[Keine/Teilweise/Komplett]. Bei der Zugriffsauswahl Teilweise können spezielle Visualisierungspläne/Planer/Trendprotokolle ausgewählt werden.



Benutzerzugriffseinstellungen

Zur Verwaltung der Benutzerzugriffseinstellungen klicken Sie auf das Symbol.



Kennwortzugriff deaktivieren – Wenn der aktive Kennwortschutz deaktiviert ist.

Visualisierung PIN-Code – Wenn das aktive Kennwort deaktiviert ist kann der Zugriff durch einen allgemeinen PIN-Code geschützt werden. Länge: 3 bis 8 Nummern.

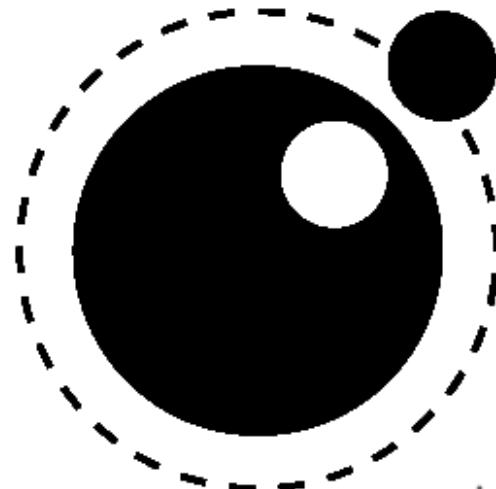


3.12 Scripting

Das Menü Scripting ermöglicht, abhängig von der Art des Skripts, das Hinzufügen und Verwalten mehrerer Skripts. Zur Implementierung der Benutzerskripts wird die Programmiersprache Lua verwendet. Die meisten Lua-Sprachaspekte werden in der ersten Ausgabe von *Programmieren mit Lua* abgedeckt, die gratis unter <http://lua.org/pil/>.

Programming in Lua

ROBERTO IERUSALIMSCHE



2nd
edition

lua.org

Last update: Wed Jan 13 12:07:35 UTC 2010

3.12.1 Ereignisbasiert

HINWEIS: Datenformat — In den meisten Fällen werden Daten zwischen spaceLYnk-Teilen mit Verwendung von hex-kodierten Strings (2 Bytes für 1 Byte an Daten) gespeichert und übertragen.

Diese Skripts werden bei Auftreten eines Gruppenereignisses am Bus ausgeführt. Wird normalerweise verwendet, wenn eine Antwort in Echtzeit erforderlich ist.

Durch Drücken des Pfeils an der unteren Seite der Schaltflächen **Ereignisbasiert, Resident oder Geplant**, erscheinen 2 Optionen:

Listenansicht – Sortiert Skripts in der Listenansicht.

Neues Script hinzufügen – Fügen Sie der Liste ein neues Script hinzu.

Beim Hinzufügen eines neuen Skripts sollten dir folgenden Felder ausgefüllt werden:

Skriptname – Der Name des Skripts.

Gruppenadresse/Tag – Die Gruppenadresse muss eingegeben werden. Tags werden in dem Dropdown gelistet.

Gruppenadressen - Ermöglicht nur die Eingabe von Ziffern von 0,9 und / als Trennzeichen. Wenn das Symbol an der rechten Seite des Textfeldes erscheint wurde eine falsche Adressform verwendet. Eine korrekte Form der Gruppenadresse ist beispielsweise 1/1/1.

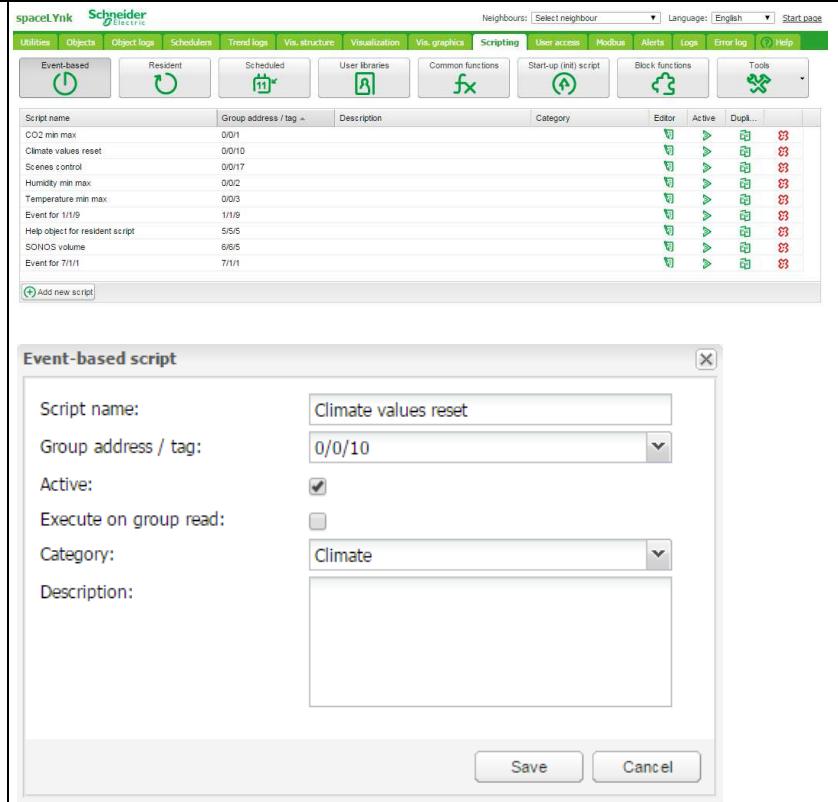
Tag – Skript kann auf Tags ausgeführt werden. Wenn Gruppenadressen Tags hinzugefügt sind und Skripts Tags verwenden, dann wird jedes Telegramm, welches mit diesem Tag an die Gruppe gesendet wird, vom Skript ausgeführt.

Aktiv – Spezifiziert, ob das Skript aktiv (grüner Kreis), oder deaktiviert (roter Kreis) ist.

Bei Gruppenlesung ausführen – Spezifiziert, ob das Skript auf einem KNX-Gruppen-Lese-Telegramm ausgeführt wird.

Kategorie – Ein neuer oder existierender Name einer Kategorie, in die das Skript eingeschlossen wird. Dies beeinflusst die Aktion des Skripts nicht, sondern hilft nur bei der Gruppierung der Skripte und der Beobachtung nach Kategorien in der Tools [Druckskript Auflistung-Seite](#).

Beschreibung – Beschreibung des Skripts.



HINWEIS: Verwenden Sie das folgende Beispiel, wenn das Skript nur bei einer Leseanforderung ausgeführt wird:

```
if event.type == 'groupread' then
    -- script here
end
```

3.12.2 Resident

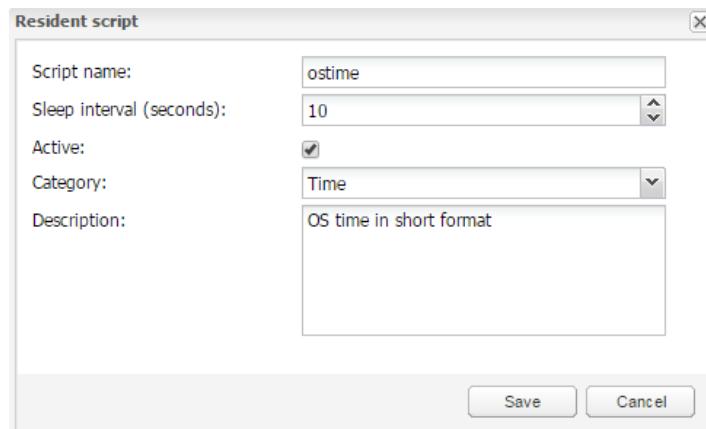
Skriptname – Der Name des Skripts.

Sleep-Intervall (Sekunden) – Das Intervall, nachdem das Skript ausgeführt wird.

Aktiv – Spezifiziert, ob das Skript aktiv (grüner Kreis), oder deaktiviert (roter Kreis) ist.

Kategorie – Ein neuer oder existierender Name einer Kategorie, in die das Skript eingeschlossen wird. Dies beeinflusst die Aktion des Skripts nicht, sondern hilft nur bei der Gruppierung der Skripte und der Beobachtung nach Kategorien in den Tools [Druckskript Auflistung-Seite](#).

Beschreibung – Beschreibung des Skripts.



3.12.3 Geplant

Skriptname – Der Name des Skripts.

Minute – Minute.

Stunde – Stunde.

Tag des Monats – Tag des Monats.

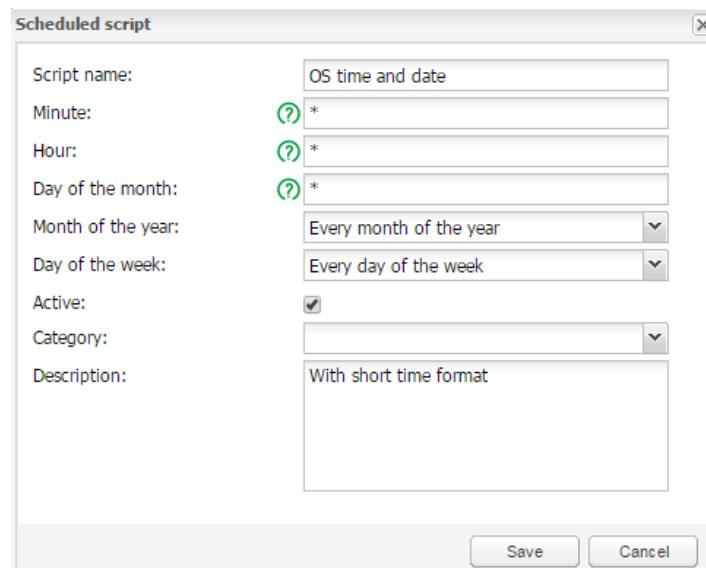
Monat im Jahr – Monat im Jahr.

Wochentag – Wochentag.

Aktiv – Spezifiziert, ob das Skript aktiv (grüner Kreis), oder deaktiviert (roter Kreis) ist.

Kategorie – Ein neuer oder existierender Name einer Kategorie, in die das Skript eingeschlossen wird. Dies beeinflusst die Aktion des Skripts nicht, sondern hilft nur bei der Gruppierung der Skripte und der Beobachtung nach Kategorien in den Tools [Druckskript Auflistung-Seite](#).

Beschreibung – Beschreibung des Skripts.



3.12.4 Benutzerbibliotheken

Benutzerbibliotheken beinhalten normalerweise benutzerdefinierte Funktionen, die später von anderen Skripts angerufen werden.

Code sichern

Es gibt eine **Option Quelle** für Benutzerbibliotheken verfügbar halten. Nach der Deaktivierung wird der Code im Binärformat erstellt und kann nicht für die weitere Bearbeitung erkannt werden. Mit aktivierter Option kann der Quellcode im Editor gesehen werden.

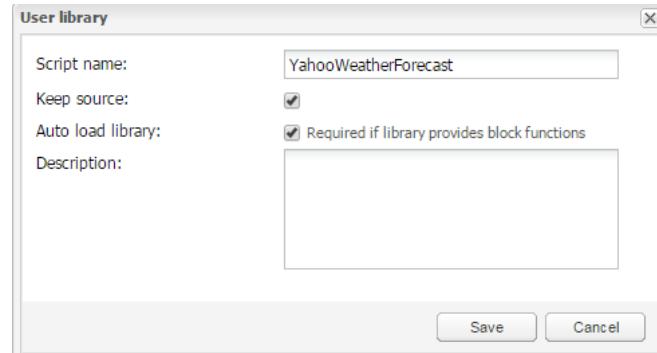
Die Option Bibliothek automatisch laden lädt das ausgewählte Script, wenn spaceLYnk gestartet wird.

Bibliothek in Skripts einbinden

Um diese, in der Benutzerbibliothek definierten, Funktionen zu verwenden, müssen diese zu Beginn des Skripts eingebunden werden. Eine Benutzerbibliothek mit dem Namen „Test“ sollte beispielsweise wie nachfolgend eingebunden werden:

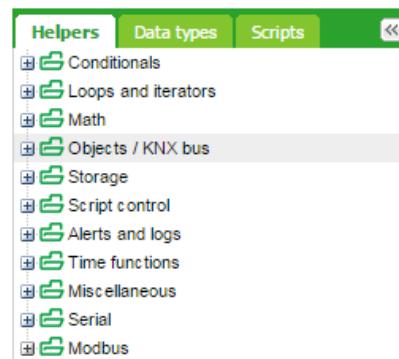
```
require('user.test')
```

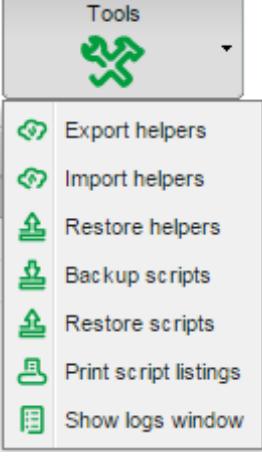
Benutzerbibliotheken können gesichert werden, und vom Archiv wiederhergestellt/hinzugefügt werden.



3.12.5 Allgemeine Funktionen

Allgemeine Funktionen enthalten eine Bibliothek mit global verwendeten Funktionen. Diese können von jedem Skript, zu jeder Zeit und ohne spezielle Einbeziehungen anderer **Benutzerbibliotheken aufgerufen werden**. Funktionen, wie Sonnenaufgang/Sonnenuntergang, E-Mail sind standardmäßig in **Allgemeine Funktionen** eingebunden.



<p>3.12.6 Start-Skript</p> <p>Das Int-Script wird zur Initialisierung auf einem spezifischen System oder Bus-Werten beim Systemstart verwendet. Das Int-Script wird nach jedem Neustart des Systems ausgeführt (Einschalten, Neustart in der SW oder per Hardware-RESET-Drucktaste).</p>	
<p>3.12.7 Tools</p> <p>Exporthelper – Export-Scripting-Helfer, speichern mit rechtem Mausklick.</p> <p>Importhelper – Import-Scripting-Helfer.</p> <p>Wiederherstellen-Helfer – Helfer zur Wiederherstellung des Standard-Scripting.</p> <p>Benutzerscripts sichern – Alle Skripts in einer *.gz-Datei sichern.</p> <p>HINWEIS: Die Skript-Sicherung sichert keine Benutzbibliotheken. Diese müssen separat gesichert werden.</p>	
<p>Skripts wiederherstellen – Skripts mit 2 Möglichkeiten von der Archivdatei (*.gz) wiederherstellen:</p> <ul style="list-style-type: none">• Bestehende Skripts entfernen und von der Sicherung importieren.• Bestehende Skripts behalten. <p>Skript-Listings ausgeben – Zeigt alle Skripts mit Codes im Listenformat und sortiert nach Kategorien.</p> <p>Protokollfenster anzeigen – Hier werden alle Protokolldaten aufgelistet. Dies ist ein dupliziertes Fenster Konfigurationen/Protokolle. Erlaubt das Debuggen eines Skripts und die parallele Prüfung der protokollierten Daten.</p>	

3.12.8 Allgemeine Scripting-Beschreibung

The screenshot shows the Schneider Electric spaceLYnk software interface. At the top, there is a navigation bar with tabs: Utilities, Objects, Object logs, Schedulers, Trend logs, Vis. structure, Visualization, Vis. graphics, Scripting (which is highlighted in green), User access, Modbus, Alerts, Logs, Error log, and Help. Below the navigation bar is a toolbar with several icons: Event-based, Resident, Scheduled, User libraries, Common functions, Start-up (init) script, Block functions, and Tools. The main area displays a table of scripts. The columns are: Script name, Group address / tag, Description, Category, Editor, Active, Dupl..., and Delete. The table contains the following data:

Script name	Group address / tag	Description	Category	Editor	Active	Dupl...	Delete
CO2 min max	0/0/1						
Climate values reset	0/0/10		Climate				
Scenes control	0/0/17						
Humidity min max	0/0/2						
Temperature min max	0/0/3						
Event for 1/1/9	1/1/9						
Help object for resident script	5/5/5						
SONOS volume	6/6/5						
Event for 7/1/1	7/1/1						

At the bottom left of the table area, there is a button labeled "Add new script".

Sie können mit jedem Scripting fünf Aktionen durchführen:

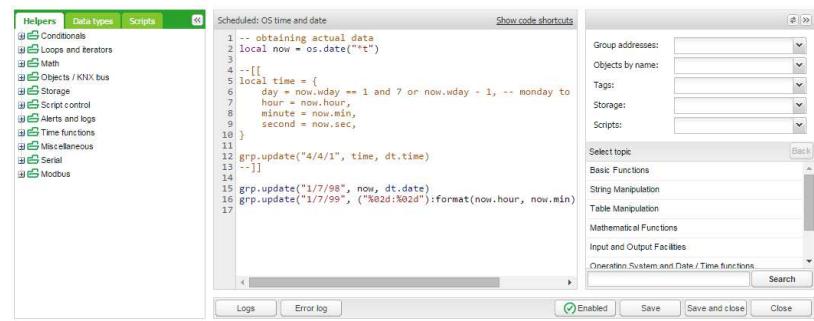
- **Duplizieren** – Duplizieren des Skripts mit Quellcode.
- **Editor** – Zugriff auf den Scripting-Editor, um einen spezifischen Code für ein bestimmtes Programm zu schreiben.
- **Aktiv** – Skript aktiv (grün) oder inaktiv (rot) setzen.
- **Bearbeiten** – Skriptname, Beschreibung, Kategorie und andere Parameter bearbeiten.
- **Löschen** – Skript löschen. Löschen muss beim Drücken auf dieses Symbol bestätigt werden.

3.12.9 Skript-Editor

Beim Hinzufügen eines Skripts erscheint ein Symbol  in der Spalte Editor, welches Ihnen das Öffnen des Skripts im Scripting-Editor und das Nachbearbeiten dieses mit integrierten Codeausschnitten ermöglicht. Codeausschnitte sparen Zeit und gestalten die Codierung komfortabler. Nach einem Klick auf den entsprechenden Snippet fügt dieser automatisch den Code in das Editor-Feld ein.

Zur Unterstützung beim Schreiben der Skripts sind Tastenkürzel implementiert.

Es sind außerdem Protokoll- und Fehlerfenster verfügbar.



Ctrl + F – Findet die Syntax in einem Code und markiert den Text in gelb.

Ctrl + G – Nachdem der Text mit Verwendung von Strg + F gefunden wurde, kann dieser mit Strg + G verwendet werden, um die nächste Syntax in einem Skript auszuwählen.

Shift + Ctrl + G – Vorherige Syntax auswählen.

Shift + Ctrl + F – Syntax im Skript durch eine andere austauschen. Sie können diese nacheinander auswählen, wenn Sie diese tauschen möchten.

Shift + Ctrl + R – Alle Syntaxen in einem Skript auf einmal durch eine andere austauschen.

Ctrl + Space – Hilft bei der automatischen Erkennung von Code und der automatischen Einfügung. Drücken Sie auf Strg + Leertaste, und tippen Sie den ersten Buchstaben des Befehls. Danach können Sie aus einer Liste den korrekten auswählen.

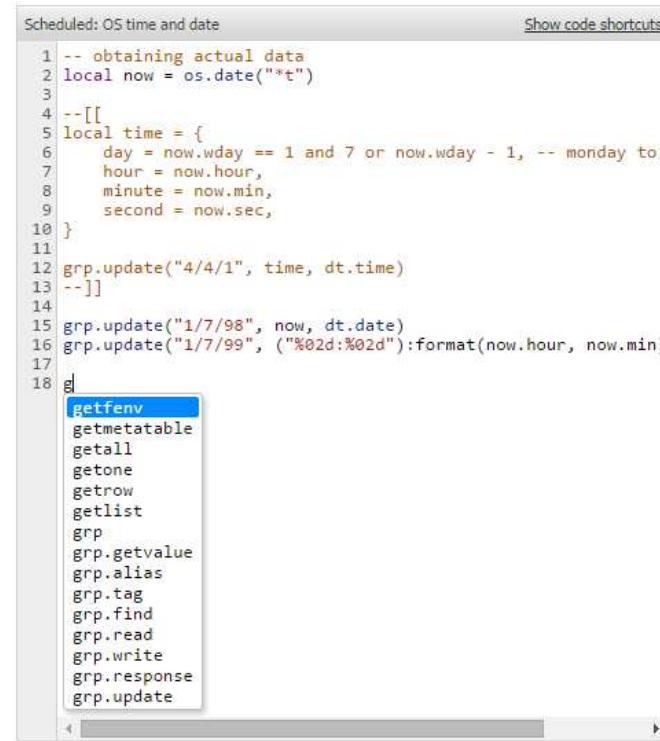
Im Skript-Editor gibt es sechs Hauptgruppen:

Helper – Vordefinierte Codeausschnitte, wie if-then Statement. Helper bestehen aus elf Haupt-Untergruppen: **Konditionelle** – If Else If, If Then, usw.

Schleifen und Iteratoren – Array, Repeat...Until, usw.

Mathematisch – Zufallswert, Obergrenze, Absoluter Wert, Runden, usw.

Objekte/KNX-Bus – Objektwert abfragen, Gruppenauslesung, Gruppe schreiben,



Aktualisierungsintervall, usw. Speicher – Daten von Speicher abfragen, Daten in Speicher speichern. Skriptsteuerung – Anderen Skriptstatus abfragen, andere Skripte aktivieren oder deaktivieren. Warnungen und Protokolle – Warnung, Protokollvariablen, Formatierte Warnung. Zeit-Funktionen – Verzögerung Skriptausführung. Diverse – Sonnenaufgang/Sonnenuntergang, usw. Seriell – Kommunikation durch interne spaceLYnk I/O-Ports. Modbus – RTU/TCP-Verbindung erstellen, Register schreiben, Register lesen, usw. Gruppenadressen – Am KNX-Bus bestehende Gruppenadressen. Objekte nach Namen – Objekt nach Namen auswählen. Tags – Objekt nach Tag auswählen. Datentypen – Objekt nach Datentyp auswählen. Skripte – Die Liste aus bereits erstellten Skripts besteht aus 4 Untergruppen: Ereignis-basierend - Liste an Ereignis-basierenden Skripts. Resident - Liste von Resident-Skripts. Geplant – Liste von geplanten Skripts. Benutzbibliotheken - Liste von allgemeinen Funktionen in Star-Up (init) Skripts.	
---	--

3.13 Warnungen

In der Registerkarte **Warnungen** befindet sich eine Liste an Warnmeldungen, in der Skripte mit Warnfunktionen definiert sind. Die Meldungen werden in der Haupt-Datenbank gespeichert.

Alert time	Script name	Message
09.02.2015 09:19:29	Event for 1/1/9	Alert: temperature is too high 26°C
09.02.2015 09:18:37	Event for 1/1/9	Alert: temperature is too high 26°C
09.02.2015 09:17:03	Event for 1/1/9	Alert: temperature is too high 26°C
09.02.2015 09:14:44	Event for 1/1/9	My alert text
09.02.2015 09:14:11	Event for 1/1/9	My alert text

Warnung (Meldung, [var1, [var2, [var3]]])

Speichert Warnmeldungen und die aktuelle Systemzeit in der Haupt-Datenbank.

Beispiel:

```
Temperatur = 25.3

wenn Temperatur > 24 dann
-- Meldung: Temperaturbereiche zu hoch: 25.3'

Warnung('Temperaturbereich zu hoch: %.1f', Temperatur)

end
```

3.14 Protokolle

Protokolle können zum Debuggen von Skriptcodes verwendet werden. Die Protokollmeldungen erscheinen, wie in der Log-Funktion definiert.

Log time	Script name	Message
09.02.2015 15:49:41	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1080.8
09.02.2015 15:49:40	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 381.4
09.02.2015 15:48:40	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1080.7
09.02.2015 15:48:40	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 379.4
09.02.2015 15:47:40	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1080
09.02.2015 15:47:40	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 377.1
09.02.2015 15:46:40	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1080.3
09.02.2015 15:46:40	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 377.1
09.02.2015 15:45:40	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1079.5
09.02.2015 15:45:40	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 375.6
09.02.2015 15:44:40	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1080.5
09.02.2015 15:44:40	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 377.6
09.02.2015 15:43:40	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1076.9
09.02.2015 15:43:40	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 364.9
09.02.2015 15:29:02	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1077.6
09.02.2015 15:29:02	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 366.2
09.02.2015 15:28:02	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1081.2
09.02.2015 15:28:02	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 385.3
09.02.2015 15:27:02	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1082.2
09.02.2015 15:27:02	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 383.8
09.02.2015 15:26:02	Temperature CPU	* arg: 1 * string: t2 * arg: 2 * number: 1082.6
09.02.2015 15:26:02	Temperature CPU	* arg: 1 * string: t1 * arg: 2 * number: 384.3

log(var1, [var2, [var3, ...]])

Wandelt Variable in eine von Personen lesbare Form um und speichert diese.

Beispiel:

```
-- Die Protokolfunktion akzeptiert Variablen vom Typ Lua nil, Boolesch, Nummern und Tabellen (mit bis zu 5 verschachtelten Ebenen).
```

```
a ={ key1 ='value1', key2 =2}
```

```
b ='test'
```

```
c =123.45
```

```
-- protokolliert alle übergebenen Variablen
```

```
log(a, b, c)
```

3.15 Fehlerprotokoll

Error time			Script name	Error description
05.02.2015 10:58:15			event-Event for 1/1/4	modbus error: Connection timed out
05.02.2015 10:58:12			event-Event for 1/1/8	modbus error: Connection timed out
05.02.2015 10:58:05			event-Event for 1/1/4	modbus error: Connection timed out

Fehlermeldungen von Skripts werden in der Registerkarte Fehlerprotokoll angezeigt.

3.16 Hilfe

spaceLYnk Schneider Electric

Utilities Objects Object logs Schedulers Trend logs Vis. structure Visualization Vis. graphics Scripting User access Modbus Alerts Logs Error log Help

Help topics

- Scripting
 - Basics and script types
 - Object functions
 - Data type functions
 - Common functions, storage, alerts
 - Scheduled scripting date/time format
 - Extended function library
 - About

Scripting basics

The main purpose of scripting is to provide custom logic functions for various KNX objects. There are three types of scripts possible:

- Event-based — executed when event occurs on a specific group address
- Resident — used for polling data from local object database
- Scheduled — executed on a specified date or time

Lua programming language is used to implement user scripts. Most of the Lua language aspects are covered in the first edition of "Programming in Lua" which is freely available online.

Notes

Data format — in most cases data is stored and transferred using hex-encoded strings (2 bytes per 1 byte of data).

Event-based scripting

Event-based scripting can be used to implement custom logic for group address events. User-defined function is executed when a "group write" event occurs for given group address. Event information is stored in global **event** variable. Variable contents:

- dst (string) — decoded destination group address (e.g. '1/1/4')
- src (string) — decoded source individual address (e.g. '1.1.2')
- type (string) — type of event, either "groupwrite", "groupresponse" or "groupread"
- datahex (string) — data as a hex-encoded string which can be converted using data type functions

event.getvalue function can be used to get the event value if the event is bound to a known object.

Note: **event** variable is available only in Event-based functions, not in Resident and Scheduled.

4 Modbus RTU und Modbus TCP

4.1 Charakteristiken

Die offene Norm Modbus ermöglicht Ihnen den Empfang einer genaueren Verbrauchsanalyse aller Gebäudeteile.

Sie können 31 Modbus-Slave-Geräte der folgenden Typen an Messgeräten, basierend auf der Modbus-Remote-Terminal-Unit (RTU) innerhalb einer Modbus-Leitung, verbinden:

- Schneider Electric Energiemesser
- Schneider Electric Leistungsmesser
- Schneider Electric Smartlink
- Modbus TCP/RTU Geräte, die nicht von Schneider Electric kommen (für mehr Flexibilität)

Mit dieser, von spaceLYnk bereitgestellten, Information können Sie Energie- oder Medienkonsum visualisieren. Dies kann außerdem zur Reduzierung von Verbrauch durch die Verwendung von Steuerstrategien innerhalb des KNX/IP-Netzwerk verwendet werden.

Modbus RTU wird über die RS485-Schnittstelle unterstützt. Modbus RTU wird über den Ethernet-Port unterstützt. Die Verbindungseinstellungen von Modbus werden mit *Verwendung der Modbus-Tabelle* im spaceLYnk- *Konfigurator durchgeführt*. Modbus-Register können mit Verwendung vordefinierter Modbus-Profile einfach zugeordnet werden.

Modbus Master kann direkt von Skripts gesteuert werden (normalerweise wird das Resident-Skript zur Auslesung von Modbus-Werten nach einem spezifischen Zeitintervall, oder dem Schreiben dieser in das KNX-Objekt oder -Visualisierung, verwendet).

Nachdem ein Skript hinzugefügt wurde, kann der Code im Skript-Editor hinzugefügt werden. In den Helfern können jede Menge vordefinierter Code-Blocks gefunden werden.

HINWEIS: Verwenden Sie die Modbus-Einstellungen mit Profilen zusammen mit von Skripts gesteuertem Modbus. Interferenz dieser beiden Einstellungen kann zu Kommunikationsfehlern führen. Wir empfehlen dringend die Verwendung von Modbus-Geräteprofilen, anstatt die Konfiguration durch Scripting.

Anwendungsbeispiel:

Anforderungen

- Messen und visualisieren Sie, wie viel Energie für die Beleuchtung eines Bürogebäudes verwendet wird.
- Messen Sie den Gas- und Wasserverbrauch des Gebäudes
- Überwachen Sie die Qualität des Netzwerks, um einen sicheren Betrieb der IT-Ausstattung zu gewährleisten.

Lösung

- Installieren Sie zur Messung des Energieverbrauchs durch Licht ein Messgerät vom Typ iEM3150.
- Installieren Sie zur Bestimmung der Qualität des Stromnetzes ein Messgerät vom Typ iEM3255.
- Verbinden Sie die Geräte miteinander über Modbus.

4.3 Modbus-Geräteprofile

Zur Zuordnung der Modbus-Adressen (Register) zu KNX-Gruppenobjekten in spaceLYnk stehen jede Menge einsatzbereite Geräteprofile zur Verfügung. Sollte die Notwendigkeit bestehen, Modbus-Register zu lesen/schreiben, so können Sie einfach Zuordnungsregeln festlegen, die Ihnen durch lesen/schreiben von KNX-Gruppenobjekten Zugriff auf das Modbus-Register ermöglichen.

Nachfolgend eine Liste der einsatzbereiten Modbus-Profile in spaceLYnk:

Modbus-Gerät	RTU-Scan	Modbus-Gerät	RTU-Scan
Compact_NSX-Compact_NSX_E	Nein	iEM-iEM3250	Ja
Masterpact_NT_NW-Masterpact_A	Nein	iEM-iEM3255	Ja
Masterpact_NT_NW-Masterpact_H	Nein	iEM-iEM3350	Ja
Masterpact_NT_NW-Masterpact_P	Nein	iEM-iEM3355	Ja
PM-PM1200	Nein	PM-PM710	Nein
PM-PM210	Nein	PM-PM750	Nein
PM-PM3250	Ja	PM-PM810	Nein
PM-PM3255	Ja	PM-PM820	Nein
PM-PM5110	Nein	PM-PM850	Nein
PM-PM5111	Nein	PM-PM870	Nein
PM-PM5310	Nein	PM-PM9C	Nein
PM-PM5330	Nein		Nein
PM-PM5350	Nein	Smartlink-RTU	Ja
iEM-iEM3150	Ja	Smartlink-TCP	Nein
iEM-iEM3155	Ja		

Mit Verwendung der Scan-Funktion können Sie Modbus-Geräte finden, die über Modbus RTU mit spaceLYnk verbunden sind. Diese Funktion finden Sie hier: [Konfigurator -> Modbus -> RTU-Scan](#).

HINWEIS: Geräte, die als „RTU-Scan = Nein“ markiert sind, können mit der automatischen Scan-Funktion nicht erkannt werden.

4.3.1 Neue Profildefinition

Sie können Ihr eigenes Profil definieren, sollte Ihr Modbus-Geräteprofil nicht in der Liste der einsatzbereiten Profile im spaceLYnk- *Konfigurator*, vorhanden sein.

Modbus-Geräteprofile werden in *.json-Dateien verteilt. Zur Erstellung oder Bearbeitung Ihres Profils können Sie einen gewöhnlichen Text-Editor, wie Notepad, verwenden. Setzen Sie die Dateierweiterung beim Speichern auf *.json. Das folgende Beispiel zeigt die Struktur eines neuen Geräteprofils:

```
{  
  "manufacturer": "Schneider Electric",  
  "description": "Example device",  
  "mapping": [  
    { "name": "Output 1", "bus_datatype": "bool", "type": "coil", "address": 0, "writable": 1 },  
    { "name": "Input 1", "bus_datatype": "float16", "type": "inputregister", "address": 0,  
      "value_multiplier": 0.001, "units": "V" }  
  ]  
}
```

Jede Zeile der Tabelle "Zuordnung" der .json-Datei enthält Informationen über die Zuordnung eines Modbus Register, Coil, Input oder Output. Alle möglichen Einstellungen für die Zuordnung sind in der nachfolgenden Tabelle aufgelistet.

Parameter	Beschreibung	Typ	Erforderlich
Name	Objektname, z.B. Output 2	String	Ja
Bus_datatype	KNX-Objekt Datentyp, Schlüssel von dt-Tabelle, z.B. float32	String / Nummer	Ja
Typ	Modbus-Registertyp, mögliche Werte: coil, discreteinput, register, inputregister.	String	Ja
Adresse	Register-Adresse (0-based)	Nummer	Ja
Beschreibbar	Zur Aktivierung von Schreiben in Register, wenn der Typ entweder „coil“ oder „discreteinput“ ist auf „Wahr“ setzen.	Boolesch	Nein
Datentyp	Modbus-Wert Datentyp. Wenn gesetzt wird die Konvertierung automatisch durchgeführt. Mögliche Werte: boolean, uint16, int16, float16, uint32, int32, float32, uint64, int64, quad10k, s10k	String	Nein
Value_delta	Der neue Wert wird gesendet, wenn der Unterschied zwischen dem vorher gesetzten Wert und dem aktuellen Wert größer als Delta ist. Standardwert ist 0 (nach jeder Auslesung senden).	Nummer	Nein
Value_multiplier	Multiplizieren Sie den Ergebniswert mit der spezifizierten Nummern - value = value_base + value * value_multiplier.	Nummer	Nein

Value_bitmask	Anzuwendende Bit-Maske. Das Verschieben wird automatisch, basieren auf der unwichtigsten, in der Maske gefundenen 1 erledigt.	Nummer	Nein
Value_nan	Anordnung von 16-Bit-Ganzzahlen. Wenn spezifiziert und der gleiche Array von der Auslesung zurückgegeben wird, wird keine weiter Verarbeitung des Werts durchgeführt.	Array	Nein
Value_conv	Übernehmen einer der integrierten Konvertierungsfunktionen	String (Intern)	Nein
Value_custom	Name der integrierten Aufzählung oder Liste an Schlüsseln -> Wertzuweisung. Wenn kein Schlüssel gefunden wird resultiert der Wert in 0.	String / Objekt	Nein
Intern	Nicht sichtbar für den Benutzer, wenn auf „Wahr“ gesetzt. Sollte für Skala-Register verwendet werden.	Boolesch	Nein
Einheiten	KNX-Objekteinheiten/Suffix	String	Nein
Address_scale	Adresse des Registers mit Wertskalierung, Wert= Wert * 10 ^ Skala	Nummer	Nein
Read_count	Anzahl der Register, die auf einmal ausgelesen werden (für Geräte, die nur das Auslesen eines spezifischen Blocks des Registers unterstützen)	Nummer	Nein
Read_swap	Registersortierung während Umwandlung tauschen (Endianness)	Boolesch	Nein
Read_offset	Position des ersten Registers an Daten vom Block der Register (0-basiert).	Nummer	Nein

Nachdem Sie Ihre .json-Datei mit all Ihren Profilinformationen erstellt haben können Sie diese einfach über **Konfigurator -> Modbus -> Profile -> Profil hinzufügen auf spaceLYnk hochladen.**

4.4 Modbus RTU-Schnittstelle

4.4.1 Charakteristiken der Modbus RTU-Schnittstelle

- Unterstützt über die physische Schnittstelle RS 485
- Kann als Modbus/RTU Master oder Modbus/RTU Slave dienen
- Unterstützte Funktionscodes: #01, #02, #03, #04, #05, #06, #07, #0F, #10
- Maximal 32 Geräte auf Bus
- Die RS 485-Schnittstelle ist nicht isoliert!

4.4.2 RS 485 Bus Topologie-Merkmale

Hauptcharakteristiken RS 485

- Betriebsmodus: Differential
- Spannung an jedem Bus-Terminal: -7 V bis +12 V
- Eingangssensitivität Empfänger: +/-200 mV
- Senk-/Quellstrom: 60 mA
- Integrierter, asymmetrischer Schutz gegen Spannungsspitzen durch elektrostatische Entladung (ESD), schnelle transiente elektrische Störgrößen (EFT), und Blitze.
- Nicht isolierte RS-485-Schnittstelle

Connection Type	<ul style="list-style-type: none">• point-to-point connections• point-to-multipoint connections
Type of Trunk Cable	shielded cable with 1 twisted pair and at least a third conductor
Maximum Length of Bus	1,000 m (3,280 ft) at 19,200 bit/s with the Telemecanique TSX CSA• cable
Maximum Number of Devices (without repeater)	32 (1 UL) devices, i.e. 31 slaves
Maximum Length of Tap Links	<ul style="list-style-type: none">• 20 m (65 ft) for one tap link• a total of 40 m (131 ft) for all tap links available on the bus

Übliche Erdungsleitung

EDs wird ein dritter Draht (in 2-Draht-Systemen) benötigt, um die Spannung zwischen Treiber und Empfänger innerhalb des zulässigen Bereichs zu halten (-7 V bis +12 V).

Dieser Draht wird als gemeinsame Schaltung verwendet und muss daher direkt an die Schutzerde, bevorzugt an einem Punkt für den kompletten Bus, angeschlossen werden.

Als Erdungspunkt für den gesamten Bus sollten Sie das Hauptgerät oder dessen Leitung wählen.

VORSICHT: Kein Terminal für den Kabelschirm. Für längere Kabel in einer rauen Umgebung empfehlen wir die Verwendung einer zusätzlichen Schirmklemme nahe dem Controller, um EMV-Störungen abzuleiten.

VORSICHT: Geerdete Verbindungen von USB, RS232, und Modbus sind miteinander verbunden. Ableitstrom kann den Bediener des Controllers verletzen.

Maximale Geräteanzahl ohne Repeater

Ein RS 485-Netzwerk kann prinzipiell aus maximal 32 Geräten bestehen.

Wenn Sie mehr als 32 Gerte mit einem Standard RS 485-Treiber verbinden möchten müssen Sie einen Repeater in Ihr Netzwerk integrieren.

Stabilisierung des Netzwerks

Ohne Datenaktivität am Modbus-Bus, wenn sich z.B. alle Knoten im Empfangsmodus befinden und kein aktiver Treiber verfügbar ist, ist der Status der Leitung unbekannt. In diesen Fällen ist die Leitung externen Geräuschen oder Interferenzen ausgesetzt. Zur Verhinderung der Aufnahme undefinierte Zustände durch den Empfänger muss die Leitung stabilisiert werden, d.h. der konstante Zustand der Leitung muss durch ein externes Widerstandspaar, welches mit dem symmetrischen Paar RS 485 verbunden ist, aufrecht erhalten werden.

RC-Terminierung

Stellen Sie zur Vermeidung unerwünschter Effekte, wie Reflexionen aus Ihrer Modbus SL-Anwendung sicher, dass Übergangsleitungen korrekt terminiert sind.
Verwenden Sie zur Minimierung von Schleifenstrom und Leitungsreflexionen eine RC-Terminierung Darüber hinaus erhöht die RC-Terminierung den Rauschabstand.
Wählen Sie zwei Serienkondensatoren von 1 nF (10 V Minimum) und zwei Widerstände von 120 Ω (0,25 W) als Leitungsterminierung. Integrieren Sie diese Komponenten an beiden Enden Ihrer Modbus SL Kommunikationsleitung.

Nur R-Terminierung

Sollte der Kunde nur auf eine R=150 Ohm-Terminierung bestehen (nicht RC), so muss er die externen Polarisierungswiderstände 450 - 650 Ohm selbst verbinden (am Master-Tap).
Siehe Schema im nachfolgenden Bild.

Modbus-Schnittstellenisolation

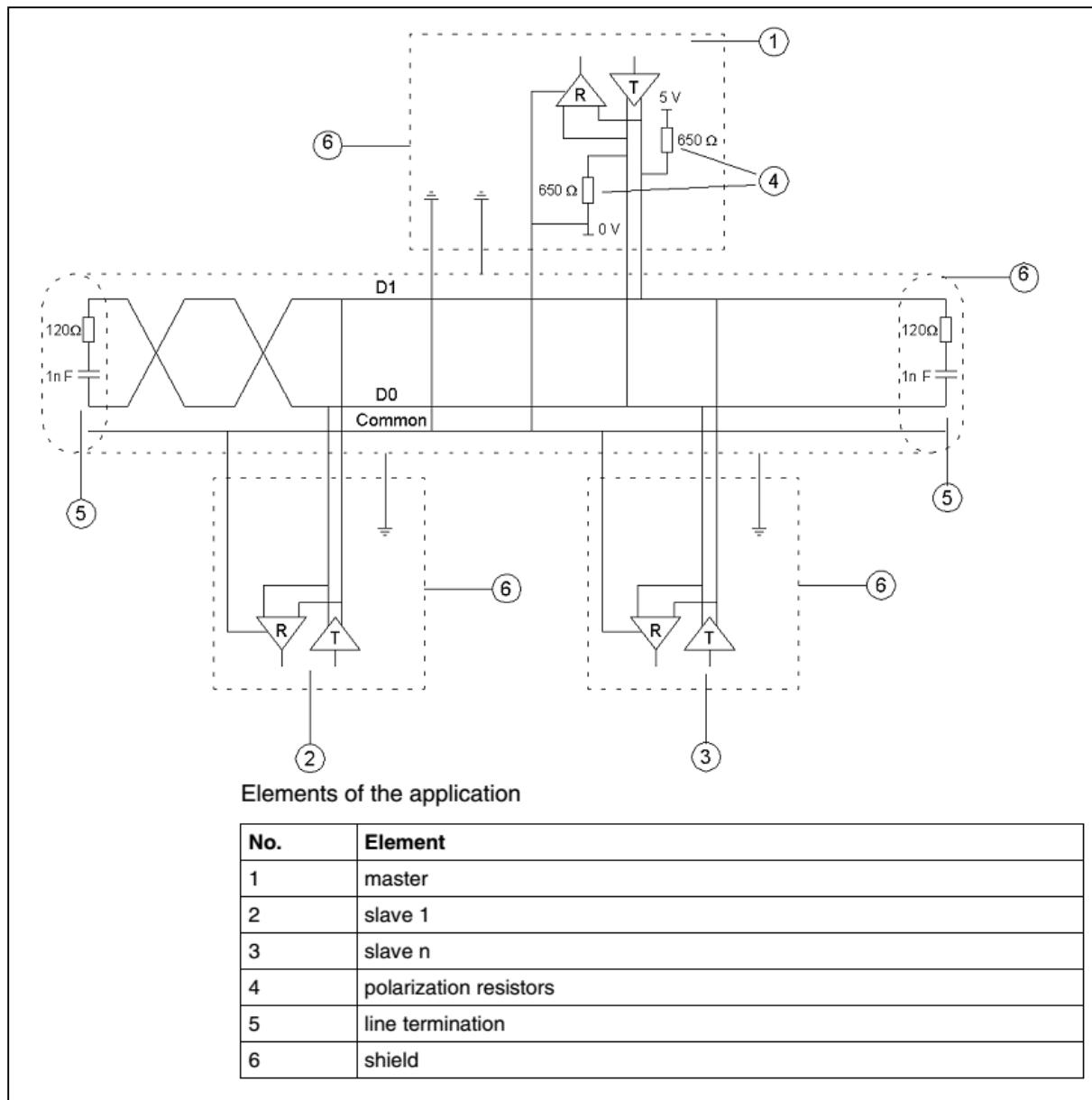
Die Modbus-Schnittstelle von spaceLYnk ist nicht isoliert. Wir empfehlen dringend die Verwendung einer Optokoppler-Trennung von Modbus-Leitung und der spaceLYnk RS485-Schnittstelle. Dies erhöht die Robustheit des Modbus-Netzwerks und reduziert Geräusche im Bus-Traffic.

Terminierung, Polarisation und Trennung mit Verwendung von Geräten von Schneider Electric

Schneider Electric hat Geräte für die RS 485-Terminierung, -Polarisation und -Trennung im Angebot. Beziehen Sie sich dafür auf die folgenden Produktnummern:

- TWD XCA ISO
- TWD XCA T3J

TWD XCA ISO und TWD XCA T3J-Geräte können zur Sicherstellung des empfohlenen RS 485-Verbindungsschemas verwendet werden (siehe nachfolgendes Bild). Detailliertere Informationen über TWD XCA ISO und TWD XCA T3J finden Sie in der Produktdokumentation auf der Website von Schneider Electric.



Maximale Kabellänge ohne Terminierung

Für ein Kabel ohne Terminierung müssen Sie die Übertragungsrate der Kabellänge anpassen.

Transmission Rate	9600	19,200	57,600	115,200
Bit Time (μ s)	104.17	52.08	17.36	8.68
Bit Time/4 (μ s)	26.0	13.0	4.3	2.2
Max. Cable Length Without Termination (m)	859	430	143	72

4.4.3 Konfigurationsbefehle Modbus RTU

[Erstellen Sie ein Modbus RTU Objekt](#)

```
require('luamodbus')
mb = luamodbus.rtu()
```

[Öffnen Sie eine Modbus RTU-Verbindung](#)

```
-- 19200 Baudrate, gerade Parität, 8 Datenbits, 1 Stoppbit, Halbduplex
mb:open('/dev/RS485', 19200, 'E', 8, 1, 'H')
mb:connect()
```

[Terminal-Name:](#)

'/dev/RS485'

[Unterstützte Baudraten:](#)

```
300 Bit/s
600 Bit/s
1200 Bit/s
2400 Bit/s
4800 Bit/s
9600 Bit/s
19200 Bit/s
38400 Bit/s
57600 Bit/s
115200Bit/s
230400Bit/s
```

[Parität:](#)

„N“ Keine
„E“ Gleich
„O“ Ungerade

[Datenbits und Stoppbits](#)

Datenbits: [Anzahl der Datenbits = 5, 6, 7, **8**]

Stoppbits: [Anzahl der Stoppbits **1**, 2]

Duplex:

- „H“ Halbduplex
- „F“ Vollduplex (in RS-485 nicht unterstützt)

Die Baudrate hängt von der Distanz zwischen den Modbus RTU-Geräten ab. Beispiel: mit einer Baudrate von 9600 Bit/s ist die maximale Kommunikationsdistanz zwischen 1 - 15 Modbus RTU-Geräten 1200 Meter. Mit einer Baudrate von 19200 Bit/s beträgt die maximale Kommunikationsdistanz 900 Meter, wie in der nachfolgenden Tabelle angezeigt:

Baudrate setting	Maximum communication distance for 1 to 15 Modbus RTU devices (Typical with Belden 3105A cables)
9600 bit/sec	1200 m
19200 bit/sec	900 m

Die Parität bezieht sich auf die Prüftechnik, wenn die Übertragung bei der Übertragung zwischen den Geräten erfolgreich war. Diese zeigt an, wenn während der Übertragung Daten verloren gegangen sind.

Einstellung der Parität

Der Modbus unterstützt nur 11 Bit-Frames. „Parität“ bezieht sich auf die Anzahl von 1s in einer gegebenen binären Zahl. Ungerade Parität bedeutet, dass eine ungerade Anzahl an 1s vorliegt. Gerade Parität bedeutet, dass eine gerade Anzahl an 1s vorliegt. Paritätsbits werden zur Fehlererkennung verwendet, wenn digitale Daten übertragen und empfangen werden.

Sowohl Gateway als auch Messgerät müssen untereinander immer gesetzt sein - ungerade, gleich, oder keine. Der standardmäßige Paritätsmodus ist „gleiche“ Parität.

- Parität = Keine: wählen Sie zwischen einem und zwei Stoppbits
- Parität = Gleich: Ein Stopbit ist festgelegt
- Parität = Ungleich: Ein Stopbit ist festgelegt

Verzögerung zwischen Frames

Einige Geräte benötigen eine erhebliche Zeit nach dem Ende einer Antwort, bis sie zum Empfang der folgenden Anforderung vom Master bereit sind. Fortan gilt dies für Schneider Electric SEPAM-Leistungsgeräte und ältere Slave-Geräte. Da diese beim Umgang mit den ursprünglichen Anforderungen langsam sind, verpassen sie die nachfolgende Anforderung.

Entsprechend der Modbus-Anforderungen sollte der Zeitraum zwischen den Anforderungen größer als 3,5 Zeichen sein. Diese älteren Geräte benötigen jedoch mehr Zeit. Bitte verwenden Sie den Befehl zur Verzögerung entsprechend:

--1,5 Sekunden warten

os.sleep(1.5)

Die Kommunikation selbst kümmert sich um einen Zeichenverzug vom minimal 3,5.

Slave-Adresse festlegen:

--Slave-Adresse auf 123 festlegen

mb:setslave(123)

[1.247]

Register auslesen:

--von Adresse 1000 auslesen und auf Wert schreiben
value = mb:readregisters(1000)

Modbus-Verbindung schließen:

mb:close()

Beispiel:

```
--Init Modbus bei erster Skriptausführung
if not mb then
    require('luamodbus')
    mb = luamodbus.rtu()
    mb:open('/dev/RS485', 38400, 'E', 8, 1, 'H')
    mb:connect()
end
mb:setslave(30)
mb:flush()
```

Zeitüberschreitungsintervall zwischen zwei aufeinanderfolgende Bytes der gleichen Meldung

```
mb:getbytettimeout()
mb:setbytettimeout(timeout)
```

Zum Warten auf die Antwort verwendetes Zeitüberschreitungsintervall:

```
mb:getresponsetimeout()
mb:setresponsetimeout(timeout)
```

Für eine eingehende Indikation vom Master verwendetes Zeitüberschreitungsintervall (nur im Slave-Modus):

```
mb:getreceivetimeout()
mb:setreceivetimeout(timeout)
```

4.5 Modbus TCP Interface

4.5.1 Charakteristiken der Modbus TCP-Schnittstelle

- Unterstützung über Ethernet-Schnittstelle 10Mb, 100Mb
- Kann als Modbus/TCP-IP Client oder Modbus/TCP-IP Server dienen
- DHCP-Unterstützung
- Max. geöffnete TCP-Verbindungen: 100
- Unterstützte Funktionscodes: #01, #02, #03, #04, #05, #06, #07, #0F, #10

Erdungsisolation:

- Metallabdeckung des RJ45-Socket ist mit der Geräteerdung grundiert.

4.5.2 Konfigurationsbefehle Modbus TCP

Erstellen Sie ein Modbus TCP-Objekt

```
require('luamodbus')
mb = luamodbus.tcp()
```

Öffnen Sie eine Modbus TCP-Verbindung

```
-- IP: 192.168.1.2, Port: 1234
mb:open('192.168.1.2', 1234)
mb:connect()
```

Der Rest der Befehle, der für die Modbus TCP-Verbindung konfiguriert werden muss, sind die gleichen als für Modbus RTU.

4.6 Modbus-Funktionscodes und entsprechende Master-Funktionen

Nachfolgend sind alle Modbus-Funktionscodes aufgelistet, die in spaceLYnk verwendet werden können. Für jeden Funktionscode steht in spaceLYnk eine Lua-Funktion zur Verfügung.

Alle nachfolgend beschriebenen Funktionen können sowohl für Modbus TCP, als auch für Modbus RTU verwendet werden.

FC#01 Read Coils:

Name „Auslesung einer einzelnen Spule“
Befehl `coil = mb:readcoils(address)`
Argumente [address]: Adresse der Spulen
Zurückgegebene Werte 1: ON, 0: OFF
Ausnahmecode: 01 oder 02 oder 03 oder 04

Name "Auslesung mehrerer Spulen"
Befehl `coil= mb:readcoils(start, count)`
Argumente [start]: Adresse der ersten Spule zur Auslesung
[count]: Anzahl der auszulesenden Spulen (max. 2000)
Zurückgegebene Werte 1: ON, 0: OFF
Ausnahmecode 01 oder 02 oder 03 oder 04
Beispiel `coil1,coil2,coil3= mb:readcoils(1000, 3)`
Der von Adresse 1000 ausgelesene Wert wird in Variable coil1 zurückgegeben.
Der von Adresse 1001 ausgelesene Wert wird in Variable coil2 zurückgegeben.
Der von Adresse 1002 ausgelesene Wert wird in Variable coil3 zurückgegeben.

FC#02 Read Discrete Inputs:

Name „Auslesung diskrete Eingaben“
Befehl `value = mb:readdiscreteinputs(address)`
Argumente [address]: Adresse der Eingabe
Zurückgegebene Werte 1: ON, 0: OFF
Ausnahmecode: 01 oder 02 oder 03 oder 04

Name „Auslesung diskrete Eingabe“
Befehl `value = mb:readdiscreteinputs(start,count)`
Argumente [address]: Adresse der ersten Eingabe zur Auslesung
[count]: Anzahl der auszulesenden Eingaben (max. 2000)
Zurückgegebene Werte 1: ON, 0: OFF
Ausnahmecode: 01 oder 02 oder 03 oder 04
Beispiel `bool1, bool2= mb: readdiscreteinputs(10,2)`
Der von der diskreten Eingabeadresse 11 ausgelesene Wert wird in Variable bool1 zurückgegeben.
Der von der diskreten Eingabeadresse 12 ausgelesene Wert wird in Variable bool2 zurückgegeben.

FC#03 Read Holding Registers:

Name „Register auslesen“
Befehl value = mb:readregisters(address,count)
Argumente [address]: Adresse des ersten Registers zur Auslesung
[count]: Anzahl der auszulesenden Register (max. 125)
Zurückgegebene Werte 2byte Werte
Ausnahmecode: 01 oder 02 oder 03 oder 04
Beispiel *int1, int2= mb: readregisters(1100,3)*
Der von Register 1100 ausgelesene Wert wird in Variable int1 zurückgegeben.
Der von Register 1101 ausgelesene Wert wird in Variable int2 zurückgegeben.
Der von Register 1102 ausgelesene Wert wird in Variable int3 zurückgegeben.

FC#04 Read Input Registers:

Name „Auslesung Eingaberegister“
Befehl value = mb:readinputregisters(address,count)
Argumente [address]: Adresse des ersten Eingaberegisters zur Auslesung
[count]: Anzahl der auszulesenden Eingaberegister (max. 125)
Zurückgegebene Werte 2byte Werte
Ausnahmecode: 01 oder 02 oder 03 oder 04
Beispiel *value1, value2, value3, value4 = mb:readinputregisters(1015,4)*
Der von Eingaberegister 1015 ausgelesene Wert wird in Variable value1 zurückgegeben.
Der von Eingaberegister 1016 ausgelesene Wert wird in Variable value2 zurückgegeben.
Der von Eingaberegister 1017 ausgelesene Wert wird in Variable value3 zurückgegeben.
Der von Eingaberegister 1018 ausgelesene Wert wird in Variable value4 zurückgegeben.

FC#05 Write Single Coil:

Name „Schreibe einzelnes Bit“
Befehl value = mb:writebits(1000, value)
[starting address, value “true” or “false”/”0”]

FC#06 Write Single Register:

Name „Schreibe einzelnes Register“
Befehl value = mb:writeregisters(1000, 123)
[address, value]

FC#07 Write Multiple Coils:

Name „Schreibe mehrere Bits“
Befehl value = mb:writebits(1000, true, false,true,...)

[address, bit value1, bit value2,..{max 1968 bits}]

FC#10 Write Multiple Registers:

Name „Schreibe mehrere Register“
Befehl **value = mb:writeregisters(1000, 123, 321,222,..)**
[address, value1, value2, ..{max 123 registers}]

Ausnahmecode

mb:readcoils(start, count)
mb:readdiscreteinputs(start, count)
mb:readregisters(start, count)
mb:readinputregisters(start, count)

Diese Befehle lesen einen oder mehrere Register/Spulen von der Startadresse aus und geben im Erfolgsfall Werte zurück. Im Fehlerfall werden drei Variablen zurückgesendet:

- Nil
- Beschreibung des Ausnahmecodes
- Ausnahmecode

Die folgende Information wird von der Modicon-Website (<http://modbus.org>) und dem Handbuch für das Modbus-Anwendungsprotokoll erhalten.

MODBUS-Ausnahmecodes

Code	Name	Bedeutung
01	Illegal Funktion	Der in der Abfrage erhaltene Funktionscode ist keine zulässige Aktion für den Server (oder Slave). Dies kann der Fall sein, da der Funktionscode nur für neuere Geräte anwendbar ist und nicht in die ausgewählte Einheit implementiert wurde. Dies kann außerdem anzeigen, dass sich der Server (oder Slave) in einem falschen Status zur Verarbeitung einer Anforderung dieses Typs befindet, zum Beispiel, da dieser nicht konfiguriert ist und zur Rückgabe von Registerwerten aufgefordert wurde.
02	Illegal Datenadresse	Die in der Abfrage erhaltene DatenAdresse ist keine zulässige Adresse für den Server (oder Slave). Insbesondere ist die Kombination von Referenznummer und Übertragungslänge ungültig. Für einen Controller mit 100 Registern generiert eine Anforderung von Offset 96 und einer Länge von 5 eine Ausnahme 02.
03	Illegaler Datenwert	Der im Feld Abfragedaten enthaltene Wert für den Server (oder Slave) ist kein zulässiger Wert. Dies zeigt einen Fehler in der Struktur des Restwerts einer komplexen Anforderung, wie eine ungültige implizierte Länge, an. Dies bedeutet jedoch NICHT, dass das zur Speicherung in einem Register übermittelte Datenelement über einen Wert außerhalb der Erwartung eines Anwendungsprogramms verfügt, da das MODBUS-Protokoll über die Bedeutung eines bestimmten Werts oder eines bestimmten Registers nicht Bescheid weiß.
04	Fehler im zugewiesenen Gerät	Beim Versuch des Servers (oder Slave), die angeforderte Aktion durchzuführen, ist ein nicht behebbarer Fehler aufgetreten. (Siehe Hinweis 1)
05	Bestätigen	Spezialisiert in Verbindung mit Programmierbefehlen. Der Server (oder Slave) hat die Anforderung akzeptiert und verarbeitet diese. Dafür wird allerdings ein langer Zeitraum benötigt. Zur Verhinderung eines Timeoutfehlers im Client (oder Master) wird diese Antwort zurückgegeben. Der Client (oder Master) kann dann zur Bestimmung, ob der Vorgang abgeschlossen ist, eine Umfrageprogramm abgeschlossen Nachricht ausgeben.
06	Beschäftigt, Zurückgewiesen-Nachricht	Spezialisierte Verwendung in Verbindung mit Programmierbefehlen. Der Server (oder Slave) ist mit der Verarbeitung eines Programmbefehls mit langer Dauer beschäftigt. Der Client (oder Master) sollte diese Nachricht später erneut senden, wenn der Server (oder Slave) wieder frei ist.
07	NAK – Negative Bestätigung	Die soeben angeforderte Programmfunction kann nicht ausgeführt werden. Umfrage zum Erhalt einer detaillierten, geräteabhängigen Fehlerinformation ausstellen. Nur gültig für Programm/Umfrage 13 und 14.
08	Paritätsfehler Speicher	Spezialisierte Verwendung in Verbindung mit den Funktionscodes 20 und 21 und Referenztyp 6, um anzugeben, dass die Übergabe einer Konsistenzprüfung durch den erweiterten Dateibereich fehlgeschlagen ist. Der Server (oder Slave) versuchte, die Aufzeichnungsdatei auszulesen, erkannte

		aber einen Paritätsfehler im Speicher. Der Client (oder Master) kann die Anforderung erneut versuchen, es kann aber ein Service an Server (oder Slave) nötig sein.
0A	Gateway-Pfad nicht verfügbar	Spezialisierte Verwendung in Verbindung mit Gateways. Zeigt an, dass das Gateway nicht in der Lage war, einen internen Kommunikationspfad vom Eingangsport zum Ausgangsport zur Verarbeitung der Anforderung zuzuordnen.
0B	Antwort Gateway-Zielgerät fehlgeschlagen	Spezialisierte Verwendung in Verbindung mit Gateways. Zeigt an, dass vom Zielgerät keine Antwort erhalten wurde. Dies bedeutet normalerweise, dass das Gerät nicht im Netzwerk vorhanden ist.

4.7 Modbus Masterfunktionen

Die nachfolgend aufgelisteten Funktionen können für den RTU-Master, oder den Modbus TCP-Client verwendet werden.

[mb:setslave\(slaveid\)](#)

Setzt die Slave-ID auf das Auslesen/Schreiben von/zu

[mb:readcoils\(start, count\) \[01\]](#)

Start – Adresse der ersten Spule zur Auslesung

Anzahl – Anzahl der auszulesenden Spulen

[mb:readdiscreteinputs\(start, count\) \[02\]](#)

Start – Adresse des ersten diskreten Eingangs zur Auslesung

Anzahl – Anzahl der auszulesenden diskreten Eingänge

[mb:readregisters\(start, count\) \[03\]](#)

Start – Adresse des ersten Holding Registers zur Auslesung

Anzahl – Anzahl der auszulesenden Holding Register

[mb:readinputregisters\(start, count\) \[04\]](#)

Start – Adresse des ersten Eingangsregisters zur Auslesung

Anzahl – Anzahl der auszulesenden Eingangsregister

gibt alle Werte bei Erfolg und Nil zurück, Fehlerbeschreibung bei Fehler

[mb:writebits\(start, v1, \[v2, \[v3, ...\]\]\) \[05\]](#)

schreibt Werte auf Spulen von der Startadresse

[mb:writeregisters\(start, v1, \[v2, \[v3, ...\]\]\) \[06\]](#)

schreibt Werte auf Register/Spulen von der Startadresse

einzelnes Schreiben wird verwendet, wenn nur ein Wert angegeben wird, ansonsten mehrfaches Schreiben

gibt alle geschriebenen Werte bei Erfolg und Nil zurück, Fehlerbeschreibung bei Fehler

[mb:reportslaveid\(\)](#)

liest interne Slave-Daten aus

gibt bei Erfolg Werte zurück

gibt Nil zurück, Fehlerbeschreibung bei Fehler

4.8 Modbus Slavefunktionen

4.8.1 Empfangsdaten vom Master

mb:receive()

empfängt Daten vom Master mit 1 Minute Timeout
gibt bei Erfolg Daten als Binärzeichen zurück
gibt Nil zurück, Fehlerbeschreibung bei Fehler

4.8.2 Modbus-Zuordnung des Slave-Geräts festlegen

mb:setmapping(coils, inputs, holding_regs, input_regs)

erstellt Speicherzuordnung für die Register mit spezifizierter Größe für jeden Typ

4.8.3 Handle-Slave

mb:handleslave()

wartet auf eine eingehende Angabe von Master und sendet, wenn notwendig, eine Antwort

4.8.4 Get-Funktionen

mb:getcoils(start, count)

mb:getdiscreteinputs(start, count)

mb:getinputregisters(start, count)

mb:registers(start, count)

bekommt einen oder mehrere Register-/Spulen-/Eingangswerte von der Zuordnung von den Startadressen
gibt bei Erfolg alle Werte zurück
gibt Nil zurück, Fehlerbeschreibung bei Fehler, Ausnahmecode wenn anwendbar

4.8.5 Set-Funktionen

mb:setcoils(start, v1, [v2, [v3, ...]])

mb:setdiscreteinputs(start, v1, [v2, [v3, ...]])

mb:setinputregisters(start, v1, [v2, [v3, ...]])

mb:registers(start, v1, [v2, [v3, ...]])

setzt Werte auf Register-/Spulenzuordnung von der Startadresse
gibt bei Erfolg „Wahr“ zurück
gibt Nil zurück, Fehlerbeschreibung bei Fehler, Ausnahmecode wenn anwendbar

4.8.6 Rückruffunktionen

[mb:setwritecoilcb\(fn\)](#)

[mb:setwriteregistercb\(fn\)](#)

legt eine Rückruffunktion für Spule/Register Schreibereignis fest
Der Rückruf sollte zwei Parameter akzeptieren - Spule/Register Adresse und Wert (Boolesch oder Nummer)
für mehrere Schreibvorgänge wird der Rückruf für jede Spule/Register separat ausgeführt
zum Entfernen eines Rückrufs Nil verwenden.

Hinweis: Weitere Details zu den Modbus-Slaveeinstellungen erhalten Sie Anwendungshinweis-Dokument

[AN_016_spaceLYnk_as_a_Modbus_slave.](#)

5 Port-Weiterleitung

5.1 Einführung

Die Port-Weiterleitung wird zum Erhalt eines Remotezugsriffs zu einem IP-Gerät im lokalen Netzwerk, wie spaceLYnk, verwendet. Die Einstellungen müssen im Netzwerkrouter vorgenommen werden. Die Port-Weiterleitung wird im Handbuch des betreffenden Routers erklärt. Bei Problemen wenden Sie sich bitte an den technischen Support des Router-Providers.

5.2 HTTP und HTTPS

spaceLYnk verwendet zwei mögliche Arten an Verbindungen.

5.2.1 HTTP

Die Standardverbindung ist durch HTTP und Port 80. HTTP ist nicht verschlüsselt und daher keine gesicherte Verbindungsart. Diese Art der Verbindung ist sicher in lokalen Netzwerken, jedoch nicht für die Verwendung einer Remote-Verbindung empfohlen. Bei Auswahl dieser Verbindung muss der Port 80 im Router mit der IP des spaceLYnk weitergeleitet werden.

Zur Verbindung von spaceLYnk mit Verwendung von Port-Weiterleitung mit HTTP-Verbindung muss folgendes in den die Adressleiste des Webbrowsers eingegeben werden:

[HTTP://IP:Port](http://IP:Port)

IP ist hierbei das IP der Internetverbindung im Haus. Diese Informationen können im Inneren des Routers gefunden werden. Sie können auch den Support Ihres Internet-Dienstanbieters kontaktieren.

5.2.2 HTTPS

HTTPS ist eine gesicherte und verschlüsselte Verbindung. Diese Art der Verbindung wird dringend bei der Verwendung einer Remote-Verbindung empfohlen. Bei der Verwendung einer gesicherten Verbindung muss der Port 443 zum Router weitergeleitet werden.

Zur Remote-Verbindung durch eine gesicherte HTTPS-Verbindung muss folgendes in den die Adressleiste des Webbrowsers eingegeben werden:

[HTTPS://IP:Port](https://IP:Port)

IP ist hierbei das IP der Internetverbindung im Haus. Diese Informationen können im Inneren des Routers gefunden werden. Sie können auch den Support Ihres Internet-Dienstanbieters kontaktieren.

HINWEIS: Bei Apple-Geräten mit OS7.0 oder höher und Verwendung einer Remote-Verbindung muss zum Erhalt eines korrekten Feedbacks in der Visualisierung (Port+1) weitergeleitet werden. Für Port 80 ist dies Feedback-Port 81. Bei Verwendung eines benutzerdefinierten Ports A müssen Sie Port A zu Port 80 von spaceLYnk und Port A + 1 zu Port 81 von spaceLYnk weiterleiten. Will der Benutzer beispielsweise einen Remote-Zugriff zur Visualisierung und verwendet dabei 1234 zum Zugriff auf sein HL, so muss er 1234 zu Port 80 von spaceLYnk und 1235 zu Port 81 von spaceLYnk weiterleiten.

HINWEIS: Sollten Sie eine andere Portnummer als die Standardports 80 und 443 verwenden, so können Sie unter *Konfigurator -> Utilities -> System -> Services -> HTTP -Server zusätzliche Ports festlegen.*

6 BACnet

6.1 Charakteristiken

BACnet ist ein Kommunikationsprotokoll für Gebäudeautomatisierung und Steuernetzwerke. Dies ist ein ASHRAE – American Society of Heating, Refrigerating and Air-Conditioning Engineers, ANSI – American National Standards Institute, und ISO – International Organization for Standardization-Protokoll

spaceLYnk wurde von BACnet Testing Laboratories (BTL) als BACnet Application Specific Controller (B-ASC) zertifiziert.

BACnet wurde entwickelt, um die Kommunikation von Gebäudeautomatisierung und Steuersystemen für Anwendungen wie Heizung, Lüftung, Klimaanlagen, Beleuchtungssteuerung, Zutrittskontrolle, Brandmeldeanlagen und zugehörigen Geräten zu ermöglichen. Das BACnet-Protokoll bietet Informationsaustausch für Geräte zur Gebäudeautomatisierung. Dies ist unabhängig vom jeweiligen Gebäudeservice, den diese Geräte durchführen.

6.2 spaceLYnk-Konfiguration

Die Verbindung von spaceLYnk und einem anderen BACnet-Gerät erfolgt über einen Ethernet Physical Layer. spaceLYnk kann hier nur als BACnet-Server fungieren. Dies bedeutet, dass spaceLYnk Daten bereitstellt, die durch ein BACnet-Client-Gerät ausgelesen werden können. Das BACnet-Client-Gerät kann dann Daten auf den Server schreiben.

Da spaceLYnk ein auf KNX basierendes Gerät ist kommt die Verbindung zum BACnet-Netzwerk von KNX-Gruppenobjekten, die dann zu BACnet exportiert werden.

6.3 Objektexport

Alle KNX-Objekte in der spaceLYnk Objektliste ([Konfigurator → objekte](#)) haben den Parameter „Export“.

Durch markieren des Kontrollkästchens „Export“ wird das spezifische KNX-Objekt in BACnet als BACnet-Objekt sichtbar.

Object...	Object name	IP...	TP...	Ev...	Data type	Current v...	Log	Export	Tags	Updated at	Se...	Vis...	Cu...
0/0/1	CO2	[]	[]	[]	09. 2 byte...	409.92 ppm	[]	[]		26.01.201...	[]	[]	[]
0/0/2	Humidity	[]	[]	[]	05.001 scale	27 % RH	[]	[]		26.01.201...	[]	[]	[]
0/0/3	Temperature	[]	[]	[]	09.001 Te...	23.34 °C	[]	[]		26.01.201...	[]	[]	[]
0/0/4	Minimal CO2	[]	[]	[]	09. 2 byte...	348.8 ppm	[]	[]		24.01.201...	[]	[]	[]
0/0/5	Maximal CO2	[]	[]	[]	09. 2 byte...	0 ppm	[]	[]		02.02.201...	[]	[]	[]
0/0/6	Minimal humidity	[]	[]	[]	09. 2 byte...	0 %RH	[]	[]		02.02.201...	[]	[]	[]
0/0/7	Maximal humidity	[]	[]	[]	09. 2 byte...	0 %RH	[]	[]		02.02.201...	[]	[]	[]
0/0/8	Minimal temper...	[]	[]	[]	09.001 Te...	0 °C	[]	[]		02.02.201...	[]	[]	[]
0/0/9	Maximal temper...	[]	[]	[]	09.001 Te...	24.22 °C	[]	[]		23.01.201...	[]	[]	[]
0/0/10	Climate values ...	[]	[]	[]	09. 2 byte...	1	[]	[]		20.01.201...	[]	[]	[]

Binärobjekte erscheinen als Binärwerte, numerische Werte als analoge Werte.

Andere Datentypen werden nicht unterstützt.

6.4 BACnet-Konfiguration

Die BACnet-Konfiguration besteht aus der Einstellung der BACnet-Serverparameter in spaceLYnk. Das BMS - Building Management System erkennt die ausgesetzten Daten.

6.4.1 Konfiguration

spaceLYnk fungiert als BACnet-Server, der in **Konfigurator → Dienstprogramme → System → Netzwerk → BACnet-Einstellungen konfiguriert werden muss.**

Server Aktiviert – BACnet-Server aktivieren/deaktivieren

Geräte-ID – BACnet Geräte-ID, die im BACnet-Netzwerk eindeutig sein muss.

Kennwort – BACnet Kennwort

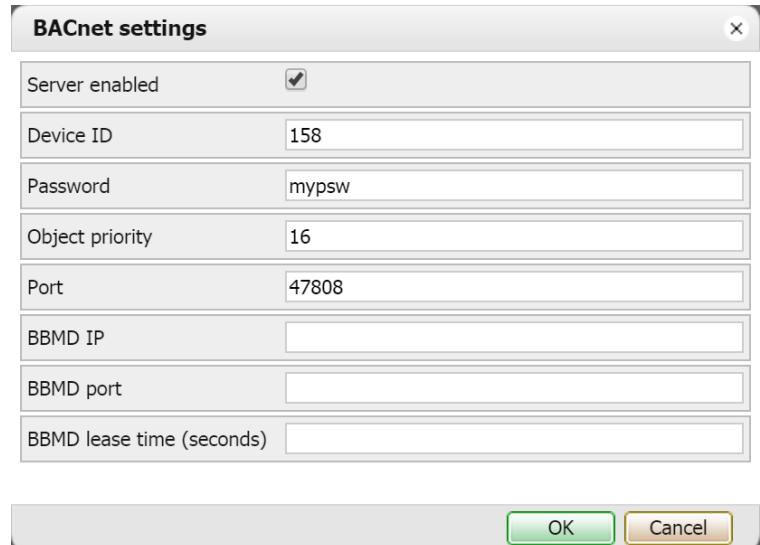
Objektpriorität – Hier definieren Sie, auf welchen Priority Array spaceLYnk schreiben wird. Standardmäßig schreibt spaceLYnk nicht auf einen Priority Array, sondern nur auf Relinquish Default Property

Port – BACnet Port, Standard 47808

BBMD IP – BACnet Router-IP.

BBMD port – BACnet Router-Port

BBMD Leasedauer (Sekunden) – Registrierung zum erneuten Senden des Intervalls



6.4.2 BACnet-Objekte

Liste an BACnet-Objekten. Deren Parameter finden Sie unter:

Konfigurator → Dienstprogramme → System → Netzwerk → BACnet-Objekte

Der Gerätename ergibt sich aus Hostname und Geräte-ID

Die BACnet-Objekte können über die Schaltfläche CSV heruntergeladen in eine CSV-Datei heruntergeladen werden.

HINWEIS: Sollte Ihr Browser diese Funktion nicht unterstützen ist die Schaltfläche CSV heruntergeladen ausgeblendet.

BACnet objects			
Type	Instance	Device name	Current value
2 (AV)	1	CO2 (0.0.1)	409.92
2 (AV)	3	Temperature (0.0.3)	23.34
2 (AV)	5	Maximal CO2 (0.0.5)	0
2 (AV)	7	Maximal humidity (0.0.7)	0
5 (BV)	256	Window 1 (0.1.0)	true
5 (BV)	257	Window 2 (0.1.1)	true
5 (BV)	258	Window 3 (0.1.2)	true
2 (AV)	1904	percents office 2 (0.7.112)	99

6.5 BACnet Standardisiertes Geräteprofil

spaceLYnk wurde in den BACnet Testing Labs (BTL) getestet und bestand alle nötigen Interoperabilitätsanforderungen.

Weitere Details und Ergebnisse betreffend des BLT-Tests können hier gefunden werden:

<http://www.bacnetinternational.net/catalog/index.php?m=20&p=1201>

6.5.1 Liste aller unterstützten BACnet Interoperability Building Blocks (BIBBs)

Datenfreigabe	ReadProperty-B	DS-RP-B
	ReadPropertyMultiple-B	DS-RPM-B
	WriteProperty-B	DS-WP-B
	COV-B	DS-COV-B
Geräte- und Netzwerkmanagement	Dynamic Device Binding-B	DM-DDB-B
	Dynamic Object Binding-B	DM-DOB-B
	DeviceCommunicationsControl-B	DM-DCC-B
	TimeSynchronization-B	DM-TS-B
	UTCTimeSynchronization-B	DM-UTC-B
	ReinitializeDevice-B	DM-RD-B

6.5.2 Unterstützte BACnet Objekttypen

- Gerät
- Analoger Wert
- Binärer Wert

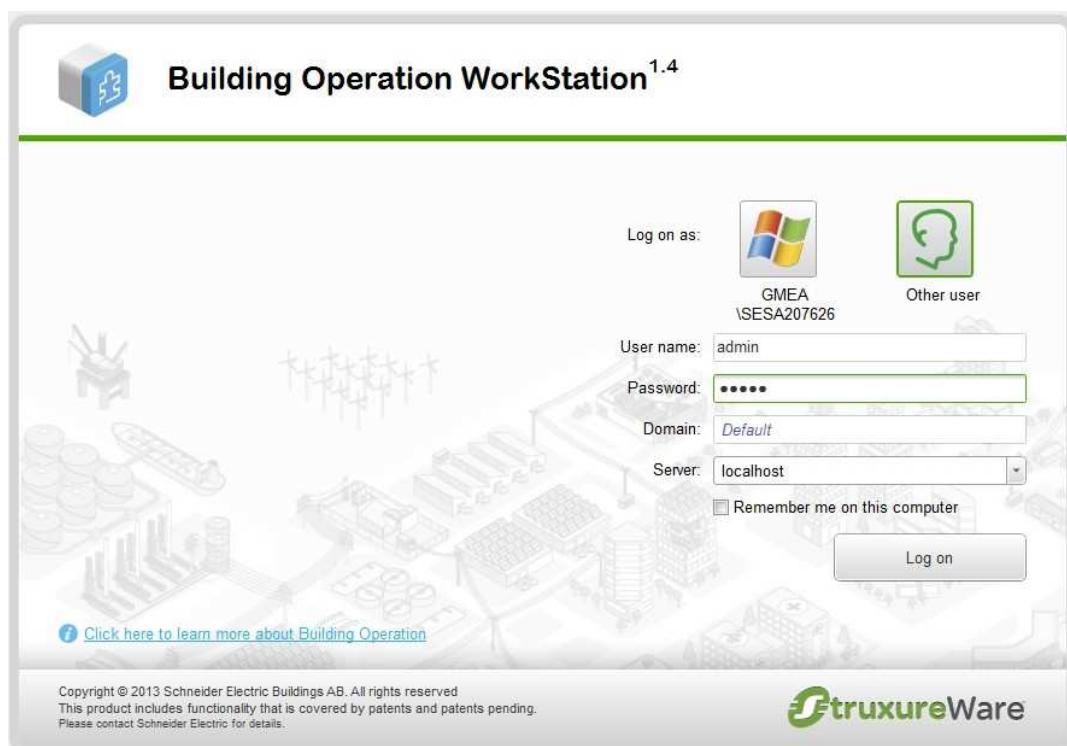
6.5.3 Layer-Optionen Datenverbindung

- Medien: BACnet IP
 - Option: Als Fremdgerät registrieren

6.6 Building Operation Workstation

Schneider Electric StruxureWare ist ein BACnet zertifiziertes Gebäudememanagementsystem. Building Operation WorkStation ist eine Software, die zur Konfiguration und Kommission eines Enterprise Server und Automation Server verwendet wird, der Daten abrufen und diese zu spaceLYnk senden kann.

Sie Interkompatibilität zwischen spaceLYnk und Building Operation Workstation über BACnet wird in einem speziellen Dokument beschrieben. Sollten Sie weitere Details über dieses Thema wünschen, beziehen Sie sich bitte auf die [Anwendungsnotiz AN001_spaceLYnk-Integration mit Verwendung von BACnet_v1.4.pdf](#).



7 RS-232 Serielle Leitung

7.1 Charakteristiken

Der Standard für serielle Schnittstellenkommunikation RS-232 wird seit vielen Jahren verwendet. Dies ist eine der am weitesten verbreiteten Verbindung zur Übermittlung von Seriendaten, da sie einfach und zuverlässig ist.

Der RS232-Serienschnittstellen-Standard ist immer noch sehr beliebt und weit verbreitet. Er wird immer noch an einigen Computern und vielen Schnittstellen gefunden und wird oft für Anwendungen zur Datenerfassung bis zur Bereitstellung einer seriellen Datenkommunikationseinrichtung in allgemeinen Computerumgebungen verwendet.

Durch die langfristige und breite Verwendung des RS232-Standard sind Produkte sowohl günstig als auch gratis verfügbar, und mit den heutigen, hohen Geschwindigkeitsstandards hat der zuverlässige und robuste RS232-Standard immer noch einiges zu bieten. Die Schnittstelle kann mit einer Reichweite von bis zu 15 Metern betrieben werden und basiert auf der ein Master/ ein Slave-Regel.

Anwendungsbeispiel:

- Verbindung an einfache Geräte oder andere Bus-Sub-Systeme.
- Audio/Video, IR-Systemintegration.

7.2 Konfigurationsbefehle

Offene Verbindung:

```
require('serial')
port = serial.open('/dev/RS232', {baudrate = 9600})
```

Schreiben Sie in den Hafen:

```
port:write('test data')
```

Auslesung Blockierung:

```
-- Skript blockiert, bis 10 Zeichen ausgelesen sind
data = port:read(10)
```

Auslesung Zeitüberschreitung:

```
-- Skript wartet für 10 Zeichen für 20 Sekunden
```

```
data = port:read(10, 20)
```

Schließen serielle Schnittstelle:

```
port:close()
```

HINWEIS: Die RS-485 serielle Leitung wird auf die gleiche Weise und mit den gleichen Konfigurationsbefehlen wie oberhalb gezeigt gesteuert. Der einzige Unterschied ist der serial.open

command:

```
port = serial.open('/dev/RS485', {baudrate = 9600})
```

HINWEIS: Weitere Informationen über die RS-232-Kommunikation erhalten Sie in der Anwendungsnotiz

[*AN010_RS232 Steuerung mit spaceLYnk_v1.0.pdf*](#)

8 USB 2.0

8.1 Charakteristiken

- USB 2.0 bietet eine Bandbreite 480 Mbit/s, entsprechend einer effektiven Bilddatenrate von 40 MB/s.
- Integrierte Spannungsversorgung (5V DC) für Geräte mit 4-poligem Kabel. Geräte, die der USB-Spezifikation entsprechen können insgesamt bis zu 500 mA vom Bus verbrauchen. Geräte mit einer Leistung von bis zu 2,5 W können daher über den Bus versorgt werden.
- Das USB-Kabel darf nur maximal 4,5 Meter lang sein.
- Übertragung kann in beide Richtungen stattfinden

Anwendungsbeispiel:

Durch Anschließen eines USB-Flashlaufwerk kann die USB-Schnittstelle zur Speichererweiterung verwendet werden.

8.2 Konfigurationsbefehle

[io.readfile \(Datei\)](#)

Komplette Datei auf einmal auslesen. Gibt Inhalte bei Erfolg als String, oder als Nil bei Fehler zurück.

[io.writefile \(Datei, Daten\)](#)

Schreibt die angegebenen Daten in eine Datei. Daten können entweder ein Wert sein, der in einen String umgewandelt werden kann, oder eine Tabelle solcher Werte. Wenn die Daten eine Tabelle sind wird jedes Tabellenelement durch ein Zeichen für eine neue Zeile beendet. Gibt einen Boolean als Schreibergebnis zurück, wenn die Datei zum Beschreiben geöffnet werden kann, oder Nil, wenn nicht auf die Datei zugegriffen werden kann.

HINWEIS:Das USB-Flashlaufwerk unterstützt das FAT, FAT32 und NTFS-Dateisystem. Die maximale Größe des Flashlaufwerks ist 32GB.

8.3 Durch Anbringen eines USB GSM-Adapters können SMS-Nachrichten gesendet und empfangen werden.

- Verwenden Sie ein Huawei E173-Modem.
- Dieses Modem muss in den USB-Port von spaceLYnk eingesteckt werden und ist sofort betriebsbereit.
- Spezifische Funktionen können mit der PIN-Code-Einstellung und Telefonnummer-White-List in die BenutzerSkript-Bibliothek eingegeben werden, die SMS-Nachrichten empfangen und versenden kann.

8.3.1 Befehlssyntax:

Zur Änderung des Objektwerts mit Verwendung von SMS, oder der zum Auslesen des Werts durch SMS-Anforderung, müssen Sie eine SMS im nachfolgenden Format senden.

Scheiben an den Bus:

- SMS-Befehlsformat: W ALIAS VALUE

Auslesen vom Bus:

- SMS-Befehlsformat: R ALIAS
- Auf Leseanfrage antwortet das Skript mit einer SMS-Nachricht, die den aktuellen Wert des ausgewählten Objekts enthält.

ALIAS kann folgendes sein:

- Gruppenadresse (z.B. 1/1/1)
- Name (z.B. Obj1). Sollte der Name Leerzeichen enthalten, so muss dieser mit Anführungszeichen abgegrenzt werden (z.B. "Room Temperature")

HINWEIS: Der Datentyp und Name des Objekts muss in **Konfigurator -> Registerkarte Objekte** gesetzt werden. Ansonsten kann das Skript nicht auf das Objekt schreiben oder dieses auslesen.

HINWEIS: In der Nachricht sind nur ASCII-Symbole zugelassen.

HINWEIS: Weitere Details über das Senden von SMS finden Sie im Anwendungshinweis-Dokument [AN011_Email SMS und FTP in spaceLYnk_v1.0.pdf](#).

9 Blockprogrammierung

9.1 Einführung

Die Programmierung von spaceLYnk kann entweder mit Lua-Skripts oder Blockprogrammierung erfolgen. Für Personen, die mit der Programmierung nicht so vertraut sind wird empfohlen, mit der Blockprogrammierung zu beginnen. Die Blockprogrammierung basiert auf Lua-Scripting, wird aber auf eine mehr benutzerfreundliche Art visualisiert.

9.2 Blockprogrammierung-Schnittstelle

Zur Erstellung von Blöcken müssen Sie diese Funktionalität aktivieren. Gehen Sie dazu zu **Dienstprogramme → Allgemeine Konfiguration → Block-Editor aktivieren**.

Sobald das Skript hinzugefügt wurde erscheint ein Puzzle-Symbol zum Zugriff auf den Block-Editor.

The screenshot shows the spaceLYnk software interface with the 'Scripting' tab selected. Below the tabs, there are six categories: Event-based, Resident, Scheduled, User libraries, Common functions, and Start-up (init) script. A table lists various scripts with columns for Script name, Group address / tag, Description, Category, Editor, Active, Dupl..., and a delete icon. The 'new script' row is currently selected.

Script name	Group address / tag	Description	Category	Editor	Active	Dupl...	
CO2 min max	0/0/1						
Climate values reset	0/0/10						
Scenes control	0/0/17						
Humidity min max	0/0/2						
Temperature min max	0/0/3						
new script	10/10/10						
Help object for resident script	5/5/5						
SONOS volume	6/6/5						

Blöcke werden auf der linken Seite nach Kategorien sortiert. Jeder Block basiert auf einem Puzzle und kann nur an entsprechender Stelle / anderen Block eingesetzt werden.

The screenshot shows the Block Editor window. At the top, it says 'Event: new script' with buttons for 'Show Lua code', 'Save as file', and 'Load from file'. On the right are 'Save', 'Save and close', and 'Close' buttons. The left sidebar lists categories: Objects, Storage, Script control, Miscellaneous, Logic, Loops, Math, Text, Lists, Variables, and Custom functions. In the center, a 'set value to' block is selected, with its properties shown: 'value' and 'Get event value'. A trash can icon is at the bottom right.

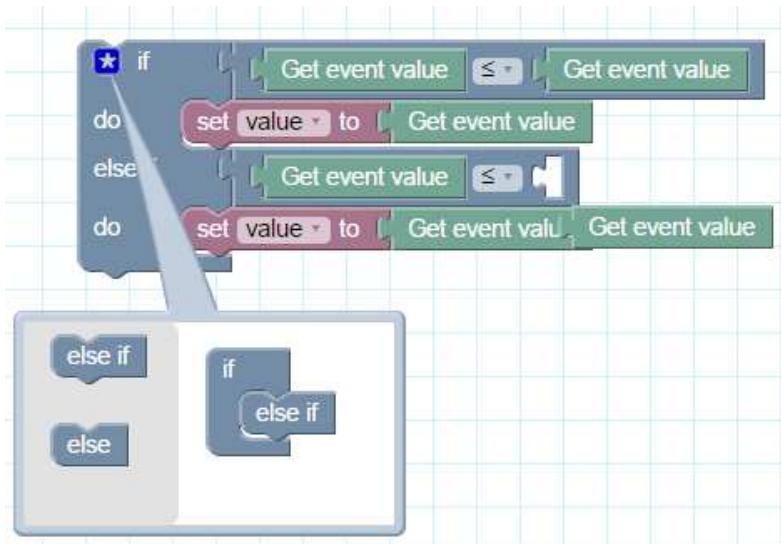
Ein Loch an der rechten Seite des Blocks markiert den Input des Blocks. Das linke Segment des Blocks markiert den Output des Blocks. Siehe nachfolgendes Beispiel:

- Der Block „set var1 to“ hat seinen Input an der rechten Seite. Dort wird der Wert ausgelesen, der in Variable var1 geschrieben wird.

- Der Block „Get current value of object 0/0/1“ hat seinen Output auf der linken Seite. Dort wird der aktuelle Wert des Objekts 0/0/1 zurückgegeben.
- Wenn die Blöcke zusammen angegeben werden, wird die Variable var1 auf den aktuellen Wert des Objekts 0/0/1 gesetzt.



Sie können die Struktur des Blocks definieren, wenn der Block mit einem Etikett an der oberen linken Ecke angezeigt wird (z.B. „if..do..else“).



Zum Löschen des Blocks klicken Sie auf die rechte Maustaste und wählen „Block löschen“ oder ziehen den Block in den Papierkorb.



Durch Klicken auf die Schaltfläche Lua anzeigen/verbergen können Sie den LUA-Code anzeigen. Dadurch können Sie die Skriptsprache lernen.

Event: Event for 1/1/1 Hide Lua code Save as file Load from file Save Save and close Close

```
1 if event.getvalue() <= event.getvalue() then
2   value = event.getvalue()
3 elseif event.getvalue() <= event.getvalue() then
4   value = event.getvalue()
5 end
6
```

HINWEIS: Das in Funktionsblöcken erstellte Skript kann in ein Lua-Skript übertragen, aber nicht wieder zurück in das Block-Funktionsskript transferiert werden. Um das Block-Funktionsskript in ein Lua-Skript zu übertragen speichern und schließen Sie Ihr Skript, und öffnen es mit Verwendung dieses Symbols im Skript-Editor .

9.3 Verwendungsbeispiele

In diesem Unterkapitel wird gezeigt, wie Funktionsblöcke korrekt verwendet werden können. Beispiele mit Beschreibungen helfen Ihnen dabei, das Konzept der Blockprogrammierung besser zu verstehen.

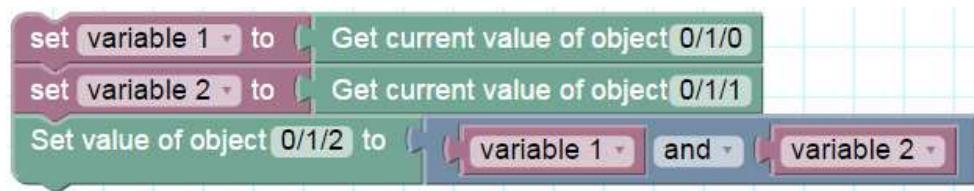
9.3.1 AND von zwei KNX-Objekten

Der aktuelle Wert von Objekt 0/1/0 wird ausgelesen und in einer Variablen mit dem Namen "variable 1" gespeichert.

Der aktuelle Wert von Objekt 0/1/1 wird ausgelesen und in einer Variablen mit dem Namen "variable 2" gespeichert.

Die logische Funktion AND wird mit den Variablen „variable 1“ und „variable 2“ durchgeführt.

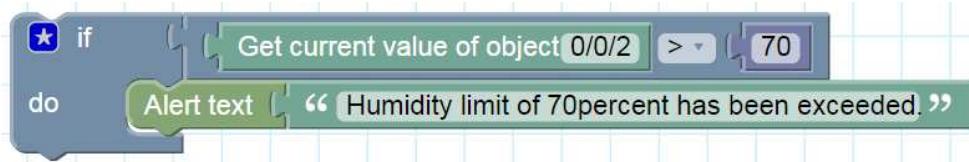
Das logische Ergebnis wird in das KNX-Objekt 0/1/2 geschrieben.



9.3.2 Eine Warnung wird erstellt, wenn der IF Objektwert das Maximum überschreitet.

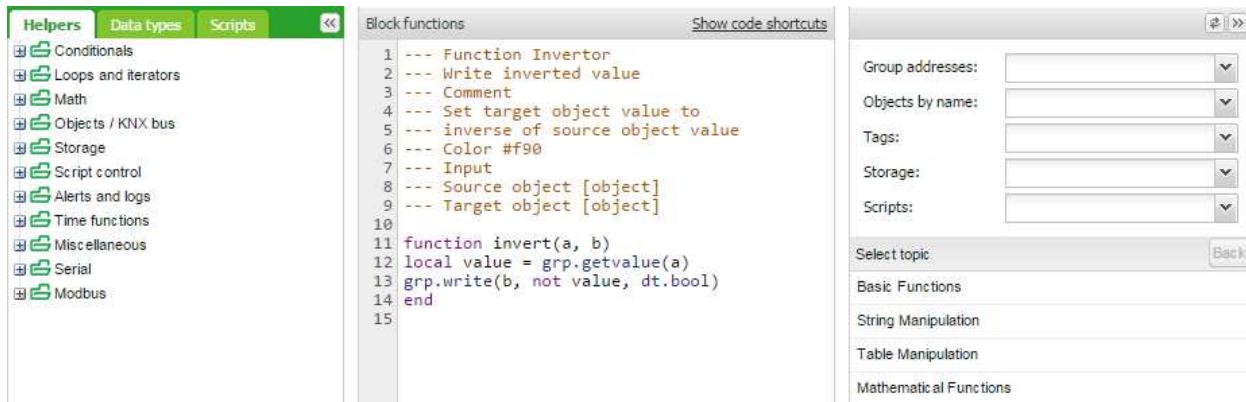
Der aktuelle Wert des Objekts 0/0/2 wird ausgelesen.

Der aktuelle Wert des Objekts 0/0/2 wird mit dem statischen Wert 70 verglichen. Eine Warnmeldung wird erstellt, wenn der Wert von Objekt 0/0/2 höher als 70 ist.



9.4 Benutzerdefinierte Funktionsblöcke

Das Menü Scripting enthält die **Schaltfläche Blockfunktionen**. Hier können Sie benutzerdefinierte Blockfunktionen erstellen, die später als fertiger Block im Block-Editor verwendet werden können.



Jede Funktion muss über einen speziellen Kommentar verfügen. Beachten Sie bitte, dass zur Erstellung der Blockfunktion in den Funktionskommentaren spezielle Schlüsselwörter (**Funktion**, **Kommentar**, **Eingabe**, optionale **Farbe**) verwendet werden müssen. Die Schlüsselwörter sind in der nachfolgenden Beschreibung markiert.

Kommentar Funktionsblock:

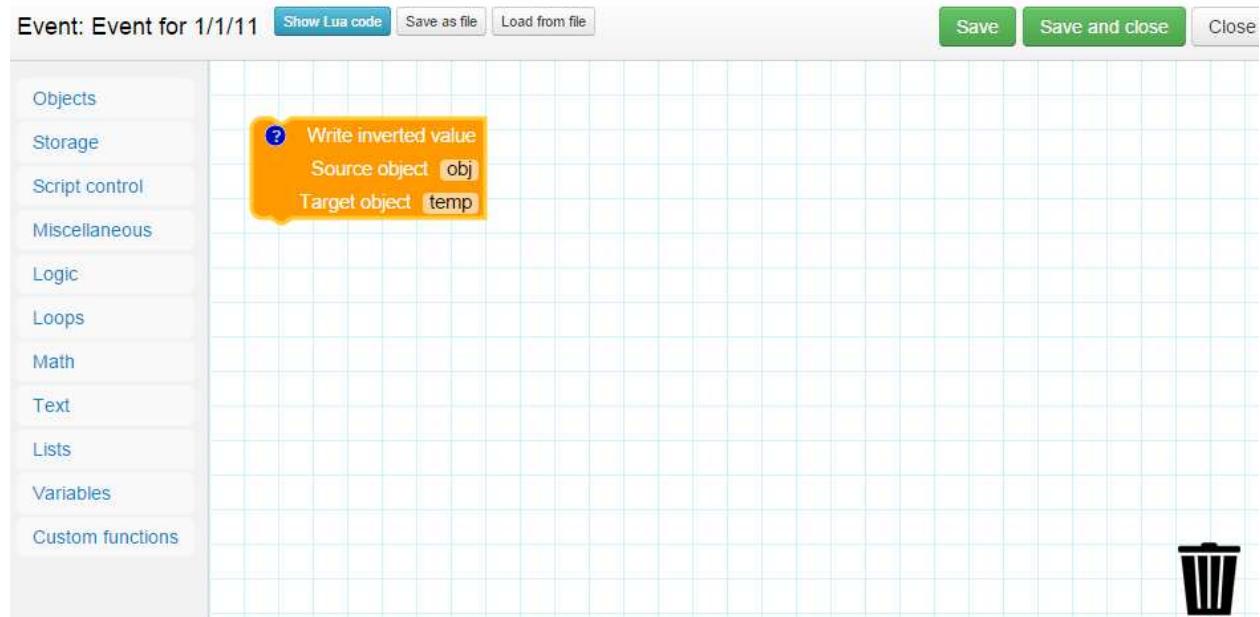
1. Die erste Zeile muss aus einem **Funktionsschlüsselwort**, gefolgt vom Funktionsnamen, bestehen.
2. Die zweite Zeile enthält eine kurze Funktionsbeschreibung, die als Blocktitel angezeigt wird.
3. Die dritte Zeile enthält das **Kommentar-Schlüsselwort**. Alle nachfolgenden Zeilen bis zur **Eingabe** werden zur Blockkommentar-QuickInfo hinzugefügt.
4. Optional kann die Blockfarbe in hexadezimal (#f00 oder #ff9900), oder numerisch als Farbtontwert zwischen 0 und 359 spezifiziert werden.
5. Die Schlüsselwort-**Eingabe** spezifiziert, dass die folgenden Zeilen die Beschreibung der Eingaben auflisten.
6. Die folgenden Zeilen enthalten Eingabelisten. Jeder Block kann über eine beliebige Anzahl an Eingaben verfügen.
Eingaben sind Funktionsparameter.
7. Wenn die Eingabedefinition in seinem Namen **[object]**, **[storage]** oder **[tag]** aufweist, so wird die Eingabe mit object, storage oder tag Auswahleingabe ersetzt.

Jede Ausgabevariable ist dem entsprechenden Rückgabewert der Funktion zugewiesen.

Beispiel:

--- **Function invert**
--- Write inverted value
--- Comment
--- Set target object value to
--- inverse of source object value
--- Color #f90
--- Input
--- Source object [**object**]
--- Target object [**object**]
function invert(a, b)
local value = grp.getvalue(a)
grp.write(b, not value, dt.bool)
end

Sobald der Block hinzugefügt wurde ist diese als Block im Block-Editor im Abschnitt Benutzerdefinierte Funktionen verfügbar.



9.5 Spezielle Funktionsblöcke

Einige spezielle Funktionsblöcke können im Abschnitt Benutzerdefinierte Funktionen gefunden werden. Diese Funktionsblöcke sind sehr einfach zu verwenden und erfordern keine Programmerkenntnisse.

Eine detailliertere Beschreibung der nachfolgend aufgelisteten Funktionsblöcke finden Sie im Anwendungshinweis-Dokument [*AN024_Block functions for spaceLYnk and homeLYnk controller.pdf*](#).

9.5.1 Allgemein

- Prozess kill (PID)
- Szene
- Schreiben Sie mit Verzögerung

9.5.2 Convertors

- Datum (Zeit) in einen String

9.5.3 Logical

- AND
- OR

9.5.4 Statistical

- AVG
- MIN
- MAX

10 LUA – Programmiersprache

LUA ist eine leistungsfähige, schnelle, leichte, einbindbare Skriptsprache. LUA kombiniert eine einfache prozedurale Syntax mit leistungsfähigen Konstrukten für Datenbeschreibung, basierend auf assoziativen Arrays und erweiterbarer Semantik. LUA ist dynamisch typisierte, wird durch einen interpretierten Byte-Code für eine registergestützte, virtuellen Maschine ausgeführt, und verfügt über eine automatische Speicherverwaltung mit inkrementellen Garbage Collection, wodurch sie ideal für Konfiguration, Scripting und Rapid Prototyping ist.

Die Programmierung in LUA als Skriptsprache für spaceLYnk basiert primär auf den Schreifunktionen. Die Funktionen im LUA **sind erstklassige Werte** mit korrektem **lexikalischen Scoping**.

Was bedeutet es für Funktionen, **erstklassige Werte zu sein**? Dies bedeutet, dass in LUA eine Funktion ein Wert mit den gleichen Rechten als ein konventioneller Wert ist, wie: Nummern und Strings. Funktionen können in Variablen (sowohl global als auch lokal) und Tabellen gespeichert werden. Diese können als Argumente übergeben, und von anderen Funktionen zurückgegeben werden.

Was bedeutet es für Funktionen, ein lexikalisches **Scoping zu haben**? Dies bedeutet, dass Funktionen auf Variablen der umgebenden Funktionen zugreifen können.

10.1 Objektfunktionen

Diese **Gruppe** bietet einen vereinfachten Zugriff auf in Datenbank gespeicherten Objekten und Gruppenadressen-Anforderungshelfer.

Die meisten **Funktionen** verwenden, alias Parameter — Objektgruppenadressen oder eindeutiger Objektname. (z.B. '1/1/1' oder 'My object')

grp.getvalue(alias)

Gibt den Wert für den angegebenen Alias zurück, oder LUA **Nil**, wenn das Objekt nicht gefunden werden kann.

grp.find(alias)

Gibt für den angegebenen Alias ein einzelnes Objekt zurück. Objektwerte werden nur dann automatisch decodiert, wenn der Datentyp im Modul „Objekte“ spezifiziert wurde. Gibt LUA **Nil** zurück, wenn kein Objekt gefunden werden kann. Gibt ansonsten eine LUA -Tabelle mit den folgenden Elementen zurück:

- **address** — Objektgruppenadresse
- **updatetime** — letzte Aktualisierungszeit im UNIX-Timestamp-Format. Zur Umwandlung in lesbare Datumsformate LUA **os.date()** verwenden.

Folgende Felder sind verfügbar, wenn der Objektdatentyp in „Objekte“ spezifiziert wurde:

- **name** — eindeutiger Objektname
- **datatype** — vom Benutzer spezifizierter Objektdatenname
- **decoded** — auf „**True**“ gesetzt, wenn decodierter Wert verfügbar
- **value** — decodierter Objektwert

grp.tag(tags, mode)

Gibt eine LUA- **Tabelle** mit Objekten mit dem angegebenen Tag zurück. Tag-Parameter können entweder eine LUA- **Tabelle**, oder ein String sein. Modusparameter können entweder „**all**“ (Rückgabeobjekte, die über alle der angegebenen Tags verfügen) oder „**any**“ (**Standard** — Rückgabeobjekte, die über einen beliebigen der angegebenen Tags verfügen) sein. Sie können **Zurückgegebenes Objekt-Funktionen** in der zurückgegebenen Tabelle verwenden.

grp.alias (alias)

Konvertiert Gruppenadressen in Objektnamen oder Namen zu Adressen. Gibt LUA **Nil** zurück, wenn kein Objekt gefunden werden kann.

10.2 Gruppenkommunikation-Funktionen

Diese Funktionen sollten nur verwendet werden, wenn ein direkter Zugriff auf Objekte durch Gruppenadressen benötigt wird. Es wird empfohlen, einzelne oder mehrere Objektfunktionen zu verwenden.

grp.write (alias, value, datatype)

Sendet Gruppen-Schreibanforderungen zum angegebenen Alias. Datenbanktyp wird von der Datenbank erhalten, wenn nicht als dritter Parameter spezifiziert. Gibt LUA **boolean** als Ergebnis zurück.

grp.response (alias, value, datatype)

Ähnlich **grp.write**. Sendet Gruppen-Antwortanforderung zum angegebenen Alias.

grp.read(alias)

Sendet Gruppen-Leseanforderungen zum angegebenen Alias.

HINWEIS: Diese Funktion gibt sofort zurück und kann nicht zur Rückgabe des Ergebnisses der Leseanforderung verwendet werden. Verwenden Sie stattdessen ein Ereignis-basiertes Skript.

grp.update(alias, value, datatype)

Ähnlich **grp.write**, sendet aber keinen Wert an den KNX TP-Bus. Senden nur Telegramme zu KNX IP wenn die KNX IP-Funktionen angeschaltet sind. Nützlich für Objekte die nur in der Visualisierung verwendet werden.

10.3 Funktionen zurückgegebener Objekte

Mit Objekten, die mit Verwendung von **grp.find(alias)** oder **grp.tag(tags, mode)** zurückgegeben wurden, sind folgende Funktionen verbunden:

Prüfen Sie immer, dass das zurückgegebene Objekt gefunden wurde. Ansonsten führt das Aufrufen dieser Funktionen zu einem Fehler. Siehe nachfolgendes Beispiel.

object:write(value, datatype)

Sendet Gruppen-Schreibanforderungen zur Gruppenadresse des Objekts. Datenbanktyp wird von der Datenbank erhalten, wenn nicht als zweiter Parameter spezifiziert. Gibt LUA **boolean** als Ergebnis zurück.

object:response(value, datatype)

Ähnlich **object:write**. Sendet Gruppen-Antwortanforderung zur Gruppenadresse des Objekts.

object:read()

Sendet Gruppen-Leseanforderungen zur Gruppenadresse des Objekts.

HINWEIS: Diese Funktion gibt sofort zurück und kann nicht zur Rückgabe des Ergebnisses der Leseanforderung verwendet werden. Verwenden Sie stattdessen ein Ereignis-basiertes Skript.

object:update(value, datatype)

Ähnlich **object:write**, sendet *aber keinen neuen Wert* an den KNX TP-Bus. Nützlich für Objekte die nur in der Visualisierung verwendet werden.

10.4 Datentyp-Funktionen

knxdatatype

Objekt bietet Codierung und Decodierung der Daten von LUA und KNX-Datenformaten.

knxdatatype.decode(value, datatype)

Konvertiert hex-codierte Daten, basieren auf dem bestimmten Datentyp, in eine LUA -Variable. Der Datentyp wird entweder als KNX primärer Datentyp (Ganzzahl zwischen 1 und 16), oder sekundärer Datentyp (Ganzzahl zwischen 1000 und 16000) spezifiziert. Rückgabewerte:

- success — decodiert aus LUA-Variable (Typ abhängig von Datentyp), Wertlänge in Bytes
- error — Nil, Fehlerstring

10.5 Datentypen

Die folgenden Datentypen können zur Codierung oder Decodierung von KNX-Daten verwendet werden. Datendarstellung auf LUA-Ebene und vordefinierte Konstanten (in fett) wie folgt:

- **bool 1 Bit (boolean) - dt.** — Boolesch
- **2 bit (1Bit gesteuert) - dt.bit2** — Nummer
- **4 Bit (3 Bit gesteuert) - dt.bit4** — Nummer
- **1 Byte ASCII-Zeichen - dt.char** — String
- **1 Byte Ganzzahl ohne Vorzeichen - dt.uint8** — Nummer
- **1 Byte Ganzzahl mit Vorzeichen - dt.int8** — Nummer
- **2 Byte Ganzzahl ohne Vorzeichen - dt.uint16** — Nummer
- **2 Byte Ganzzahl mit Vorzeichen - dt.int16** — Nummer
- **2 Byte Fließkomma - dt.float16** — Nummer
- **3 Byte Ganzzahl ohne Vorzeichen** — 232.600 RGB-Farbe
- **3 Byte Zeit / Tag - dt.time** — Tabelle mit folgenden Elementen:
 - Tag — Nummer (0-7)
 - Stunde — Nummer (0-23)
 - Minute — Nummer (0-59)
 - Sekunde — Nummer (0-59)
- **3 Byte Datum - dt.date** — Tabelle mit folgenden Elementen:
 - Tag — Nummer (1-31)
 - Monat — Nummer (1-12)
 - Jahr — Nummer (1990-2089)

- **4 Byte Ganzzahl ohne Vorzeichen - dt.uint32** — Nummer
- **4 Byte Ganzzahl mit Vorzeichen - dt.int32** — Nummer
- **4 Byte Fließkomma - dt.float32** — Nummer
- **4 Byte Zugriffssteuerung - dt.access** — Nummer, momentan nicht komplett unterstützt.
- **14 Byte ASCII String - dt.string** — String, Null Zeichen ("\"0") werden während der Decodierung verworfen

10.6 Funktionen Datenspeicherung

Speicherobjekt bietet einen anhaltenden Schlüssel-Wert-Datenspeicher für Benutzerskripte. Nur die folgenden LUA-Datentypen werden unterstützt:

- **boolesch**
- **nummer**
- **string**
- **table**

storage.set(key, value)

Setzt für den bestimmten Schlüssel einen neuen Wert. Der alte Wert wird überschrieben. Gibt Boolesch als das Ergebnis und einen optionale Fehlerstring zurück.

storage.get(key, default)

Erhält Wert für den angegebenen Schlüssel oder gibt Standardwert zurück (**Nil** wenn nicht spezifiziert), wenn Schlüssel im Datenspeicher nicht gefunden wird.

HINWEIS: Alle Benutzerskripts verwenden den gleichen Datenspeicher. Vergewissern Sie sich, dass bei der Speicherung verschiedener Arten an Daten nicht die gleichen Schlüssel verwendet werden.

Beispiel:

- Die folgenden Beispiele zeigen die grundlegende Syntax von **storage.set**. Das Ergebnis gibt boolean **true** zurück, da die übergebenen Parameter korrekt sind.

```
result=storage.set('my_stored_value_1', 12.21)
```

- Dieses Beispiel gibt **False** als Ergebnis zurück, da wir versuchen, eine Funktion zu speichern, die nicht möglich ist.

```
testfn=function(t)
return t * t
end
result =storage.set('my_stored_value_2', testfn)-- dies führt zu einem Fehler
```

- Die folgenden Beispiele zeigen die grundlegende Syntax von **storage.get**. Angenommen, der Schlüsselwert wurde nicht gefunden. Der erste Aufruf gibt **Nil** zurück, während der zweite Aufruf die Zahl **0** zurückgibt, die als Standardwert spezifiziert wurde.

```
result =storage.get('my_stored_value_3')-- gibt Nil zurück, wenn der Wert nicht gefunden wurde
result =storage.get('my_stored_value_3', 0)-- gibt 0 zurück, wenn der Wert gefunden wurde
```

- Versichern Sie sich bei der Speicherung von Tabellen, dass die zurückgegebenen Ergebnistypen prüfen. Nehmen wir an, wir haben ein Speicherelement mit dem Schlüssel **test_object_data** erstellt.

```
objectdata={}
objectdata.temperature=23.1
objectdata.scene='default'
result =storage.set('test_object_data', objectdata)-- store objectdata variable as
'test_object_data'
```

- Jetzt werden wir Daten aus dem Speicher abgerufen. Datentyp wird auf Korrektheit überprüft.

```
objectdata=storage.get('test_object_data')
if type(objectdata)=='table' then
if objectdata.temperature> 24 then
-- do something if temperature level is too high
end
end
```

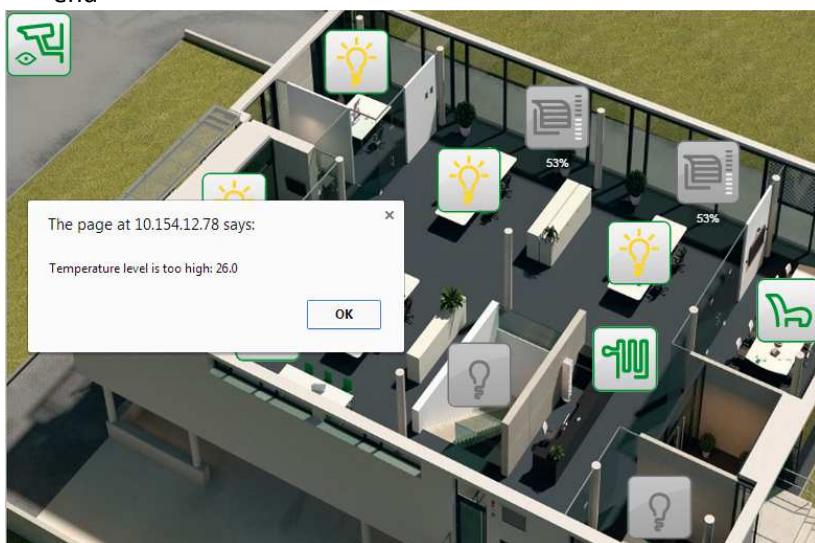
10.7 Warnfunktionen

Warnung (message, [var1, [var2, [var3]]])

Speichert Warnmeldungen und die aktuelle Systemzeit in der Haupt-Datenbank. Im Modul „Warnungen“ kann auf alle Warnungen zugegriffen werden. Diese Funktion verhält sich genau gleich wie LUA **string.format**.

Beispiel:

```
Temperatur = 25,3
wenn Temperatur > 24 dann
-- Meldung: Temperaturbereiche zu hoch: 25.3'
Warnung('Temperaturbereich zu hoch: %.1f', Temperatur)
end
```



10.8 Protokoll-Funktionen

Protokoll (var1, [var2, [var3, ...]])

Wandelt Variablen in eine von Personen lesbare Form um und speichert diese in der Haupt-Datenbank. Im Modul „Protokolle“ kann auf alle Elemente zugegriffen werden.

Beispiel:

```
-- Die Protokollfunktion akzeptiert Variablen vom Typ Lua nil, Boolesch, Nummern und Tabellen  
(mit bis zu 5 verschachtelten Ebenen).  
a ={ key1 ='value1', key2 =2}  
b ='test'  
c =123.45  
-- protokolliert alle übergebenen Variablen  
log(a, b, c)
```

10.9 Zeit-Funktionen

os.sleep(delay)

Verzögerung der nächsten Befehlsausführung für die **Verzögerungs**-Sekunden.

os.microtime()

Gibt zwei Werte zurück: aktuellen Timestamp in Senkungen und Timestamp-Bruchteile in Nanosekunden.

os.udifftime (sec, usec)

Gibt Zeitunterschiede als Fließkommawert zwischen jetzt und Timpstamp-Komponenten zurück, die zu dieser Funktion (Sekunden, Nanosekunden) übergeben wurden.

10.10 String-Funktionen

Die Bibliothek bietet generische Funktionen zur String-Manipulation, wie das Auffinden und Extrahieren von Teilstrings und Mustererkennung. Bei der Indizierung einer Zeichenfolge in LUA ist das erste Zeichen in Position 1 (nicht bei 0, wie in C).

Indizes können negativ sein und werden vom Ende des Strings als Rückwärtsindexierung interpretiert. Daher ist das letzte Zeichen bei Position -1, usw.

Der String-Bibliothek stellt alle seine Funktionen in der Tabelle String bereit. Setzt außerdem eine MetaTable für Strings, bei denen das Feld __index auf die Tabelle String zeigt. Deshalb können Sie die String-Funktionen im objektorientierten Stil verwenden. string.byte(s, i) kann beispielsweise als s:byte(i) geschrieben werden. Die String-Bibliothek übernimmt die Ein-Byte-Zeichenkodierung.

string.trim (str)

Trimmt die führenden und nachgestellte Leerzeichen aus dem bestimmten String.

string.split (str, sep)

Teilt String gegeben durch das bestimmte Trennzeichen. Gibt eine LUA-Tabelle zurück.

string.byte (s [, i [, j]])

Gibt die internen, numerischen Codes der Zeichen s[i], s[i+1], …, s[j] zurück. Der Standardwert für i ist 1; der Standardwert für j ist i.

HINWEIS: , dass Zahlencodes nicht unbedingt über Plattformen hinweg portabel sind.

string.char (…)

Empfängt Null oder mehr Ganzahlen. Gibt einen String mit einer Länge gleich der Anzahl an Argumenten zurück, in denen jedes Zeichen einen internen Zahlencode gleich seinem betreffenden Argument hat.

HINWEIS: Zahlencodes nicht unbedingt über Plattformen hinweg portabel sind.

string.find (s, pattern [, init [, plain]])

Sucht nach der ersten Übereinstimmung von Mustern im String s. Wird eine Übereinstimmung gefunden, so gibt der Fund die Indizien von s zurück, wo das Ereignis beginnt und endet. Ansonsten wird **Nil** zurückgegeben. Eine dritte, optionale numerische Argument Init. spezifiziert, wo mit der Suche begonnen wird. Der Standardwert ist 1 und kann negativ sein. Der Wert „true“ als vierter, optionales Argument Plain schaltet den Mustervergleich-Einrichtungen ab, damit die Funktion eine einfache „Teilstring finden“-Operation durchführen kann, bei der keine Zeichen im Muster als „magisch“ angesehen werden.

HINWEIS: Wenn Plain angegeben wird, muss Init auch angegeben werden. Sollte das Muster über Erfassungen verfügen, so wird der erfasste Wert nach zwei Indizes bei einer erfolgreichen Übereinstimmung auch zurückgegeben

string.format (formatstring, …)

Gibt eine formatierte Version seiner variablen Anzahl an Argumenten nach der im ersten Argument festgelegten Beschreibung zurück (welche ein String sein muss). Der Format-String befolgt die gleichen Regeln als die printf-Familie der Standard-C-Funktionen. Die einzigen Unterschiede sind, dass die Optionen/Modifizierer *, I, L, n, p, und h nicht unterstützt werden und keine extra Option q vorhanden ist. Die Option q formatiert einen String in einer Form, die sicher durch den LUA-Interpreter zurückgelesen werden kann: der String wird zwischen doppelte Anführungszeichen geschrieben, und alle doppelten Anführungszeichen, Zeilenumbrüche, eingebettete Nullen und umgekehrte Schrägstriche im String werden beim Schreiben korrekt umgeschrieben. Beispiel. Der Aufruf:

```
string.format('%q', 'a string with "quotes" and \n new line')
```

gibt folgenden String aus:

*"a string with \"quotes\" and *

new line"

Die Optionen c, d, E, e, f, g, G, i, o, u, X, und x erwarten alle eine Zahl als Argument, während, q und s einen String erwarten. Diese Option akzeptiert keine String-Werte mit eingebetteten Nullen, mit Ausnahmen als Argument zur Option q.

string.gmatch (s, pattern)

Gibt eine Iterator-Funktion zurück, die bei jedem Aufruf die nächsten Erfassungen vom Muster über String s zurückgibt. Die komplette Übereinstimmung wird in jedem Aufruf erzeugt, wenn das Muster keine Erfassungen spezifiziert. Nehmen wir folgende Schleife als Beispiel:

```
s = "hello world from LUA"
for w in string.gmatch(s, "%a+") do
    print(w)
end
```

durchläuft alle Wörter von String s und gibt einen pro Zeile aus. Das nächste Beispiel sammelt alle Paare Schlüssel=Wert aus dem angegebenen String in eine Tabelle:

```
t = {}
s = "from=world, to=LUA"
for k, v in string.gmatch(s, "(%w+)=(%w+)") do
    t[k] = v
end
```

In dieser Funktion funktioniert das „^“ am Beginn des Musters nicht als Anker, da dies die Iteration verhindern würde.

string.gsub (s, pattern, repl [, n])

Gibt eine Kopie von s zurück, in der alle (oder das erste n, wenn vorhanden) Vorkommen des Musters durch einen in repl spezifizierten Ersatz-String ersetzt werden. Dies kann ein String, eine Tabelle, oder eine Funktion sein. gsub gibt, als zweiter Wert, die totale Anzahl der aufgetretenen Übereinstimmungen zurück.

Wenn repl ein String ist, wird sein Wert als Ersatz verwendet. Das Zeichen % funktioniert als Escapezeichen: jegliche Sequenz in repl der Form %n und n zwischen 1 und 9 steht für den Wert des n. erfassten Teilstrings (siehe nachfolgend). Die Sequenz %0 steht für die komplette Übereinstimmung. Die Sequenz %% steht für ein einfaches %.

Wenn **repl** eine Tabelle ist wird für jede Übereinstimmung die Tabelle abgefragt. Dabei wird die erste Erfassung als Schlüssel verwendet; wenn das Muster keine Erfassungen spezifiziert wird die komplette Übereinstimmung als Schlüssel verwendet

Wenn **repl** eine Funktion ist, wird diese jedes Mal aufgerufen, wenn eine Übereinstimmung eintritt und deshalb mit allen Teilstrings als Argumente übergeben; wenn das Muster keine Erfassungen spezifiziert wird die komplette Übereinstimmung als Einzelargument übergeben.

Wenn der Wert durch eine Tabellenabfrage oder durch die Funktion Abfrage ist ein String oder eine Nummer, dann wird dieser als Ersatz-String verwendet; bei **Falsch** oder **Null** erfolgt kein Ersatz (hier wird die Original-Übereinstimmung im String beibehalten).

Beispiel:

```
x = string.gsub("hello world", "(%w+)", "%1 %1")
--> x="hello hello world world"

x = string.gsub("hello world", "%w+", "%0 %0", 1)
--> x="hello hello world"

x = string.gsub("hello world from LUA", "(%w+)%s*(%w+)", "%2 %1")
--> x="world hello LUA from"

x = string.gsub("home = $HOME, user = $USER", "%$(%w+)", os.getenv)
--> x="home = /home/roberto, user = roberto"

x = string.gsub("4+5 = $return 4+5$", "%$(-)%$", function (s)
    return loadstring(s)()
end)
--> x="4+5 = 9"

local t = {name="LUA", version="5.1"}
x = string.gsub("$name-$version.tar.gz", "%$(%w+)", t)
--> x="LUA-5.1.tar.gz"
```

string.len (s)

Empfängt einen String und gibt dessen Länge zurück. Der leere String „“ hat eine Länge von 0. Eingebettete Nullen werden gezählt, somit hat „a\000bc\000“ eine Länge von 5.

string.lower (s)

Empfängt einen String und gibt eine Kopie dieses Strings zurück, in dem alle Großbuchstaben nach Kleinbuchstaben geändert wurden. Alle anderen Zeichen bleiben unberührt. Die Definition welcher Großbuchstaben hängt vom aktuellen Gebietsschema ab.

string.match (s, pattern [, init])

Sucht nach der ersten Übereinstimmung von Mustern im String s. Wird eine Übereinstimmung gefunden, so gibt die Übereinstimmung die Erfassungen vom Muster zurück. Ansonsten wird **Nil** zurückgegeben. Die komplette Übereinstimmung wird zurückgegeben, wenn das Muster keine Erfassungen spezifiziert. Eine dritte, optionale numerische Argument Init. spezifiziert, wo mit der Suche begonnen wird. Der Standardwert ist 1 und kann negativ sein.

string.rep (s, n)

Gibt einen String zurück, der die Konzentration von n Kopien des Strings s ist.

string.reverse (s)

Gibt einen String zurück, der die Umkehrung von String s ist.

[string.sub \(s, i \[, j\]\)](#)

Gibt einen Teilstring von s zurück, der mit i beginnt und bis j fortfährt; i und j können negativ sein. Wenn j nicht vorhanden ist, wird angenommen, dass j gleich -1 ist (was das Gleiche wie die Länge des Strings ist). Insbesondere gibt der Aufruf string.sub(s,1,j) ein Präfix von s mit der Länge j zurück, und string.sub(s, -i) gibt einen Suffix von s mit der Länge i zurück.

[string.upper \(s\)](#)

Empfängt einen String und gibt eine Kopie dieses Strings zurück, in dem alle Kleinbuchstaben nach Großbuchstaben geändert wurden. Alle anderen Zeichen bleiben unberührt. Die Definition, was ein Kleinbuchstabe ist, hängt vom aktuellen Gebietsschema ab.

Muster

Zeichenklasse

Eine Zeichenklasse wird zur Darstellung einer Reihe an Zeichen verwendet. Zur Beschreibung einer Zeichenklasse sind die folgenden Kombinationen erlaubt:

- **x:** (wobei x keiner der Sonderzeichen ist $\^{\$}()%.[]*+-?$) steht für das Zeichen x selbst.
- **.**: (ein Punkt) steht für alle Zeichen.
- **%a:** stellt alle Buchstaben dar.
- **%c:** stellt alle Steuerzeichen dar.
- **%d:** stellt alle Ziffern dar.
- **%l:** stellt alle Kleinbuchstaben dar.
- **%p:** stellt alle Interpunktionszeichen dar.
- **%s:** stellt alle Leerzeichen dar.
- **%u:** stellt alle Großbuchstaben dar.
- **%w:** stellt alle alphanumerischen Zeichen dar.
- **%x:** stellt alle Hexadezimalziffern dar.
- **%z:** stellt alle Zeichen mit Darstellung 0 dar.
- **%x:** (wobei x ein nicht-alphanumerisches Zeichen ist) stellt das Zeichen x dar. Dies ist der normale Weg zur Vermeidung von magischen Zeichen (Sonderzeichen). Jedes Interpunktionszeichen (auch die nicht-magischen) kann bei der Verwendung zur eigenen Darstellung in einem Muster durch ein vorangestelltes '%' angezeigt werden.
- **[set]:** steht für die Klasse, welche die Vereinigung aller Zeichen im Satz ist. Ein Zeichenbereich kann durch Trennung der Endzeichen des Bereichs mit einem „-“ spezifiziert werden. Alle oben beschriebenen Klassen %x können auch als Komponenten im Satz verwendet werden. Alle anderen Zeichen im Satz stellen sich selbst dar. [%w_] (oder [_%w]) stellt beispielsweise alle alphanumerischen Zeichen plus den Unterstrich dar, [0-7] stellt alle oktalen Zahlen und [0-7%l%-] alle oktalen Zahlen, plus Kleinbuchstaben, plus das „-“-Zeichen dar.
- Das Zusammenspiel zwischen den Bereichen und Klassen ist nicht definiert. Deshalb haben Muster wie [%a-z] oder [a-%%] keine Bedeutung.
- **[^set]:** stellt die Ergänzung eines Satzes dar, wobei der Satz darüber interpretiert wird.

Für alle Klassen, die mit einzelnen Buchstaben (%a, %c, etc.) dargestellt werden, stellt der entsprechende Großbuchstabe die Ergänzung der Klasse dar. %S stellt beispielsweise alle Nicht-Leerzeichen dar.

Die Definitionen der Buchstaben, Leerzeichen und anderen Zeichengruppen hängt vom aktuellen Gebietsschema ab. Insbesondere ist die Klasse [a-z] nicht gleichbedeutend mit %l.

Musterelement:

Ein Musterelement kann folgendes sein:

- eine einzelne Zeichenklasse, die mit jeden einzelnen Zeichen in der Klasse übereinstimmt;
- eine einzelne Zeichenklasse, gefolgt von „*“, die mit 0 oder weiteren Wiederholungen der Zeichen in der Klasse übereinstimmt. Diese Wiederholungselemente stimmen immer mit der am Längsten möglichen Sequenz überein;
- eine einzelne Zeichenklasse, gefolgt von „*“, die mit 1 oder weiteren Wiederholungen der Zeichen in der Klasse übereinstimmt. Diese Wiederholungselemente stimmen immer mit der am Längsten möglichen Sequenz überein;
- eine einzelne Zeichenklasse, gefolgt von „-“, die auch mit 0 oder weiteren Wiederholungen der Zeichen in der Klasse übereinstimmt. Im Gegensatz zu „*“ stimmen diese Wiederholungselemente immer mit der am Kürzesten möglichen Sequenz überein;
- eine einzelne Zeichenklasse, gefolgt von „?“, die mit 0 oder 1 Auftreten eines Zeichens in der Klasse übereinstimmt.
- %n, für n zwischen 1 und 9; ein solches Element stimmt mit einem Teilstring gleich dem n. erfassten String überein (siehe nachfolgend);
- %bxy, wobei x und y zwei unterschiedliche Zeichen sind; ein solches Element stimmt mit Strings überein, die mit einem x beginnen, einem y enden, und bei dem x und y ausbalanciert sind. Wenn daher jemand den String von links nach rechts liest und +1 für ein x und -1 für ein y zählt, ist das Ende von y das Erste y dort, wo die Zählung 0 erreicht. Das Element %b() stimmt beispielsweise mit Ausdrücken mit ausbalancierten Klammern überein.

Muster:

Ein Muster ist eine Sequenz an Musterelementen. Ein „^“ am Beginn eines Musters verankert die Übereinstimmung am Anfang des Betreff-Strings. Ein „\$“ am Ende eines Musters verankert die Übereinstimmung am Ende des Betreff-Strings. An anderen Positionen haben „^“ und „\$“ keine spezielle Bedeutung und stellen sich selbst dar.

Erfassungen:

Ein Muster kann Teilmuster beinhalten, die umklammert sind. Sie beschreiben Erfassungen. Bei einer erfolgreichen Übereinstimmung werden die Teilstrings, die mit den Erfassungen übereinstimmen, zur zukünftigen Verwendung gespeichert (erfasst). Erfassungen werden gemäß deren linken Klammern nummeriert. Für das Muster „(a*(.)%w(%s*))“ wird beispielsweise der Teil der übereinstimmenden Strings „a*(.)%w(%s*)“ als erste Erfassung (und deshalb mit der Nummer 1) gespeichert; das mit „.“ übereinstimmende Zeichen wird mit der Nummer 2 erfasst, und der mit „%s*“ übereinstimmende Teil erhält die Nummer 3.

Als spezieller Fall erfasst () die aktuelle String-Position (eine Nummer). Wenn wir beispielsweise das Muster „()aa()“ an String „flaaap“ anwenden, resultiert dies in zwei Erfassungen: 3 und 5. Ein Muster kann keine eingebetteten Nullen enthalten. Verwenden Sie stattdessen %.

10.11 Eingabe- und Ausgabefunktionen

io.exists (Pfad)

Prüft, ob der gegebene Pfad (Datei oder Verzeichnis) existiert. Booleschen Wert zurückgeben.

io.readfile (Datei)

Liest eine komplette Datei auf einmal. Gibt Inhalte bei Erfolg als String, oder als Nil bei Fehler zurück.

io.writefile (Datei, Daten)

Schreibt die angegebenen Daten in eine Datei. Daten können entweder ein Wert sein, der in einen String umgewandelt werden kann, oder eine Tabelle solcher Werte. Wenn die Daten eine Tabelle sind wird jedes Tabellenelement durch ein Zeichen für eine neue Zeile beendet. Gibt einen Boolean als Schreibergebnis zurück, wenn die Datei zum Beschreiben geöffnet werden kann, oder Nil, wenn nicht auf die Datei zugegriffen werden kann.

10.12 Funktion Skriptsteuerung

script.enable('scriptname')

Aktiviert das Skript mit dem Namen *scriptname*.

script.disable('scriptname')

Deaktiviert das Skript mit dem Namen *scriptname*.

status = script.status('scriptname')

Gibt true/false zurück, wenn das Skript gefunden wird. Gibt ansonsten Nil zurück.

10.13 Konvertierungen

Kompatibilitätsebene: Imcore ist ein Alias von cnv.

cnv.strtohex (str)

Wandelt den angegebenen binären String in einen hex-kodierten String um.

cnv.hextostr (hex [, keepnulls])

Wandelt den angegebenen hex-kodierten String in einen binären String um. NULL-Zeichen werden standardmäßig ignoriert, können aber durch das Setzen des zweiten Parameters auf true eingebunden werden.

[cnv.tonumber \(value\)](#)

Wandelt den angegebenen Wert unter Verwendung der folgenden Regeln in eine Nummer um:
Nummern und gültige numerische Strings werden neutral behandelt, boolesch *true* ist 1,
boolesch *falsch* ist 0, alles andere ist *Nil*.

[cnv.hextoint\(hexvalue, bytes\)](#)

Wandelt den angegebenen Hex-String in eine Ganzzahl der angegebenen Länge in Bytes um.

[cnv.inttohex\(intvalue, bytes\)](#)

Wandelt die angegebene Ganzzahl in eine Hex-String der angegebenen Bytes um.

[cnv.strtohex\(str\)](#)

Wandelt den angegebenen binären String in einen hex-kodierten String um.

[cnv.hextostr\(hexstr\)](#)

Wandelt den angegebenen hex-kodierten String in einen binären String um.

10.14 Bit-Operatoren

[bit.bnot \(value\)](#)

Binär nicht

[bit.band \(x1 \[, x2...\]\)](#)

Binär und zwischen einer beliebigen Anzahl an Variablen

[bit.bor \(x1 \[, x2...\]\)](#)

Binär und zwischen einer beliebigen Anzahl an Variablen

[bit.bxor \(x1 \[, x2...\]\)](#)

Binär und zwischen einer beliebigen Anzahl an Variablen

[bit.lshift \(value, shift\)](#)

Linke binäre Verschiebung

[bit.rshift \(value, shift\)](#)

Rechte binäre Verschiebung

10.15 Eingabe- und Ausgabeeinrichtungen

Die I/O Bibliothek bietet zwei unterschiedliche Formate für die Dateibearbeitung an. das erste verwendet implizite Dateideskriptoren; das heißt, es liegen Operationen vor, um eine Standard-Eingabedatei festzulegen und alle Eingabe-/Ausgabeoperationen erfolgen über diese Standarddateien. Die zweite Form verwendet explizite Dateideskriptoren.

Bei der Verwendung expliziter Dateideskriptoren werden alle Operationen durch die Tabelle io geliefert. Bei der Verwendung expliziter Dateideskriptoren gibt die Operation io.open einen Dateideskriptor zurück, und alle Operationen werden als Methoden von Dateideskriptor geliefert.

Die Tabelle io bietet außerdem 3 vordefinierte Dateideskriptoren mit deren üblichen Bedeutungen von C: io.stdin, io.stdout und io.stderr. Diese Dateien werden von der I/O-Bibliothek niemals geschlossen.
Sofern nicht anders angegeben geben alle I/O-Funktionen bei einem Fehler Nil zurück (plus einer Fehlermeldung als zweites Resultat und einen systemabhängigen Fehlercode als drittes Resultat), und bei Erfolg einige andere Werte als Nil.

io.close ([file])

Entsprechend file:close(). Ohne eine Datei wird die Standard-Ausgabedatei geschlossen.

io.flush ()

Entsprechend file:flush über der Standard-Ausgabedatei.

io.input ([file])

Öffnet bei Aufruf mit einem Dateinamen die benannte Datei (im Text-Modus) und setzt die Handle als Standard-Eingabedatei. Beim Aufruf mit einer Dateihandle wird einfach diese Dateihandle als Standard-Eingabedatei festgelegt. Beim Aufruf ohne Parameter wird die aktuelle Standard-Eingabedatei zurückgegeben. Im Falle eines Fehlers ruft diese Funktion den Fehler hervor, anstatt einen Fehlercode zurückzugeben.

io.lines ([filename])

Öffnet den bestimmten Dateinamen im Lesemodus und gibt eine Iterator-Funktion zurück, die bei jedem Aufruf eine neue Zeile von der Datei zurückgibt. Daher durchläuft die Konstruktion alle Zeilen der Datei. Sobald die Iterator-Funktion das Ende der Datei erkennt gibt sie Nil zurück (um die Schleife zu beenden) und schließt die Datei automatisch.

for line in io.lines(filename) do body end

Der Aufruf io.lines() (ohne Dateiname) ist entsprechend io.input():lines(); dies bedeutet, dass die Zeilen der Standard-Eingabedatei durchlaufen werden. In diesem Fall wird die Datei nach Beendigung der Schleife nicht geschlossen.

io.open (filename [, mode])

Diese Funktion öffnet eine Datei mit dem in String-Modus definierten Modus. Sie gibt eine neue Dateihandle, oder im Falle eines Fehlers Nil, plus eine Fehlermeldung, zurück. Der Modus-String kann einer der folgenden sein:

- "r": Lesemodus (der Standard);
- "w": Schreibmodus;
- "a": Anfügemodus;
- "r+": Updatemodus, alle vorherigen Daten werden beibehalten;
- "w+": Updatemodus, alle vorherigen Daten werden gelöscht;
- "a+": Updatemodus anhängen, vorherige Daten werden beibehalten; Schreiben ist nur am Ende der Datei möglich.

Der Modus-String kann außerdem ein „b“ am Ende haben, was von einigen Systemen zum Öffnen der Datei im Binär-Modus benötigt wird. Dieser String ist exakt, was in der Standard-C Funktion fopen verwendet wird.

io.output ([file])

Ähnlich io.input, operiert aber über der Standard-Ausgabedatei.

10.16 Mathematische Funktionen

Diese Bibliothek ist eine Schnittstelle zur Standard-C-Mathematikbibliothek. Sie bietet alle Funktionen im Inneren der Tabelle math.

math.abs (x)

Gibt den absoluten Wert von x zurück.

math.acos (x)

Gibt den Arcuscosinus von x (in Bogenmaß) zurück.

math.asin (x)

Gibt den Arcussinus von x (in Bogenmaß) zurück.

math.atan (x)

Gibt die inverse Tangente von x (in Bogenmaß) zurück.

math.atan2 (y, x)

Gibt die inverse Tangente von y/x (in Bogenmaß) zurück, verwendet aber die Zeichen beider Parameter, um den Quadranten des Ergebnisses zu finden. (Handhabt außerdem den Fall, dass x Zero ist korrekt.)

math.ceil (x)

Gibt den kleinsten Integer, der größer oder gleich als x ist, zurück.

math.cos (x)

Gibt den Cosinus von x zurück (in Radiant angenommen).

math.cosh (x)

Gibt den hyperbolischen Cosinus von x zurück.

math.deg (x)

Gibt den Winkel von x (im Bogenmaß bestimmt) in Grad zurück.

math.exp (x)

Gibt den Wert zurück e^x .

math.floor (x)

Gibt den größten Integer, der kleiner oder gleich als x ist, zurück.

math.fmod (x, y)

Gibt den Rest der Division von x durch y zurück der den Quotienten gegen Null runden.

[math.frexp \(x\)](#)

Gibt m und e so zurück, dass $x = m \cdot 2^e$, e ist eine Ganzzahl und der absolute Wert von m ist im Bereich [0,5, 1) (oder Null, wenn x Null ist).

[math.huge](#)

Der Wert HUGE_VAL ist ein Wert, der größer oder gleich jedes anderen numerischen Werts ist.

[math.ldexp \(m, e\)](#)

Gibt zurück $m \cdot 2^e$, (e sollte eine Ganzzahl sein).

[math.log \(x\)](#)

Gibt den natürlichen Logarithmus von x zurück.

[math.log10 \(x\)](#)

Gibt den basis-10 Logarithmus von x zurück.

[math.max \(x, ...\)](#)

Gibt den maximalen Wert unter seinen Argumenten zurück.

[math.min \(x, ...\)](#)

Gibt den minimalen Wert unter seinen Argumenten zurück.

[math.modf \(x\)](#)

Gibt zwei Zahlen zurück, den integralen Bestandteil von x und den Bruchteil von x.

[math.pi](#)

Der Wert von pi.

[math.pow \(x, y\)](#)

Gibt x^y zurück. (Zur Berechnung dieses Werts können Sie außerdem den Ausdruck x^y verwenden.)

[math.rad \(x\)](#)

Gibt den Winkel von x (in Grad bestimmt) in Bogenmaß zurück.

[math.random \(\[m \[, n\]\]\)](#)

Diese Funktion ist eine Schnittstelle zur einfachen Pseudo-Zufallsgenerator-Funktion rand von ANSI C. (Es wird keine Gewähr für die statistischen Eigenschaften gegeben.)

Gibt beim Aufruf ohne Argumente eine einheitliche Pseudozufallszahl im Bereich [0,1] aus. Beim Aufruf mit der Ganzzahl m gibt math.random eine einheitliche Pseudozufallszahl in dem Bereich [1, m] zurück. Beim Aufruf mit zwei Ganzzahlen m und n gibt math.random eine einheitliche Pseudozufallszahl in dem Bereich [m, n] zurück.

[math.randomseed \(x\)](#)

Sets x als „seed“ für den Pseudo-Zufallsgenerator: gleiche seeds produzieren gleiche Zahlenfolgen.

[math.sin \(x\)](#)

Gibt den Sinus von x zurück (in Radian angenommen).

[math.sinh \(x\)](#)

Gibt den hyperbolischen Sinus von x zurück.

math.sqrt (x)

Gibt die Quadratwurzel von x zurück. (Zur Berechnung dieses Werts können Sie außerdem den Ausdruck $x^{0,5}$ verwenden.)

math.tan (x)

Gibt die Tangente von x zurück (in Radiant angenommen).

math.tanh (x)

Gibt die hyperbolische Tangente von x zurück.

10.17 Tabellenbearbeitung

Diese Bibliothek bietet generische Funktionen zur Tabellenbearbeitung. Sie bietet alle Funktionen im Inneren der Tabelle. Die meisten Funktionen in der Tabelle Bibliothek nehmen an, dass diese Tabelle einen Array oder eine Liste darstellt. Wenn wir bei diesen Funktionen über die „Länge“ einer Tabelle sprechen, meinen wir das Ergebnis des Längen-Operators.

table.concat (table [, sep [, i [, j]]])

Ein Array, bei dem alle Elemente Strings oder Nummern sind, gibt `table[i]..sep..table[i+1]` zurück `sep..table[j]`. Der Standardwert für `sep` ist der leere String, der Standard für `i` ist 1, und der Standard für `j` ist die Länge der Tabelle. Wenn `i` größer als `J` ist, wird ein leerer String zurückgegeben.

table.insert (table, [pos,] value)

Fügt bei Position `pos` in der Tabelle Elementwert ein und verschiebt, wenn nötig, andere Elemente, im Platz zu öffnen. Der Standardwert für `pos` ist `n+1`, wobei `n` die Länge der Tabelle ist, sodass ein Aufruf von `table.insert(t,x)` am Ende von Tabelle `t`, `x` einfügt.

table.maxn (table)

Gibt den größten positiven, numerischen Index der bestimmten Tabelle, oder Null zurück, wenn die Tabelle über keine positiven, numerischen Indizes verfügt. (Zur Durchführung dieses Jobs führt diese Funktion einen linearen Durchlauf der gesamten Tabelle durch).

table.remove (table [, pos])

Entfernt das Element an Position `pos` aus der Tabelle und verschiebt, wenn nötig, andere Elemente nach unten, um den Raum zu schließen. Gibt den Wert des entfernten Elements zurück. Der Standardwert für `pos` ist `n+1`, wobei `n` die Länge der Tabelle ist, sodass ein Aufruf von `table.remove(t)` das letzte Element von Tabelle `t` entfernt.

table.sort (table [, comp])

Sortiert Tabellenelemente in der bestimmten Reihenfolge, in-place, von `table[1]` zu `table[n]`, wobei `n` die Länge der Tabelle ist. Wenn `comp` bestimmt ist, muss diese eine Funktion sein, die zwei Tabellenelemente empfängt, und `true` zurücksendet, wenn die Erste weniger als die Letzte ist (so dass `not comp(a[i+1],a[i])` nach der Sortierung `true` ist). Wenn `comp` nicht bestimmt ist, wird anstatt der Standard-LUA-Operator `<` verwendet.

Der Sortieralgorithmus ist nicht stabil; das heißt, Elemente, die von der bestimmten Sortierung als gleich angesehen werden, können durch die Sortierung eine Änderung ihrer relativen Position erfahren.

10.18 Betriebssystem-Einrichtungen

os.date ([format [, time]])

Gibt einen String, oder eine Tabelle mit Datum und Zeit, formatiert in Übereinstimmung mit dem bestimmten String-Format, zurück. Bei vorhandenem Zeit-Argument ist dies die Zeit, die formatiert werden muss (siehe os.time-Funktion für eine Beschreibung dieses Werts). Ansonsten formatiert das Datum die aktuelle Zeit.

Wenn das Format mit '!' beginnt ist das Datum in Coordinated Universal Time (koordinierte Weltzeit) formatiert. Wenn das Format nach diesem optionalen Zeichen der String "*t" ist, gibt das Datum eine Tabelle mit den folgenden Feldern zurück: year (vier Zahlen), month (1--12), day (1--31), hour (0--23), min (0--59), sec (0--61), wday (Wochentag, Sonntag ist 1), yday (Tag des Jahres), und isdst (Kennzeichnung für Sommerzeit, boolesch).

Wenn das Format nicht „*t“ ist, wird das Datum als String zurückgegeben und nach den gleichen Regeln als die C-Funktion strftime formatiert.

Beim Aufruf ohne Argumente gibt das Datum eine Darstellung von Datum und Zeit zurück, die vom Host-System und dem aktuellen Gebietsschema abhängig ist (d.h., os.date() ist gleich dem os.date(„%c“)).

os.difftime (t2, t1)

Gibt die Anzahl der Sekunden von Zeit t1 bis Zeit t2 zurück. Dieser Wert ist in POSIX, Windows, und einigen anderen Systemen genau t2-t1.

os.execute ([command])

Diese Funktion ist gleich der C-Funktion System. Sie gibt den Befehl weiter, der von einer Shell des Betriebssystems ausgeführt werden muss. Sie gibt einen systemabhängigen Status-Code zurück. Wenn der Befehl nicht anwesend ist wird bei verfügbarer Shell eine ungleich Null ausgegeben, ansonsten eine Null.

os.exit ([code])

Ruft, optional mit einem Code, die C-Funktion Beenden zur Beendigung des Host-Programms auf. Der Standardwert für den Code ist der Erfolgscode.

os.getenv (varname)

Gibt den Wert der Prozessumgebungs-Variable varname zurück, oder Nil, wenn die Variable nicht definiert ist.

os.remove (filename)

Löscht die Datei oder das Verzeichnis mit diesem Namen. Zum Löschen eines Verzeichnisses muss dieses leer sein. Wenn die Funktion fehlschlägt wird Nil mit einem String über die Beschreibung des Fehlers zurückgegeben.

os.rename (oldname, newname)

Benennt die Datei oder das Verzeichnis mit dem Namen oldname in newname um. Wenn die Funktion fehlschlägt wird Nil mit einem String über die Beschreibung des Fehlers zurückgegeben.

os.time ([table])

Gibt beim Aufruf ohne Argumente die aktuelle Zeit zurück, oder eine Zeit, die das Datum und die Zeit darstellt, die durch die bestimmte Tabelle spezifiziert wird. Diese Tabelle muss über die Felder year, month, und day, und kann über die Felder hour, min, sec, und isdst verfügen (eine Beschreibung dieser Felder finden Sie in der Funktion os.date).

Der zurückgegebene Wert ist eine Zahl. Die Bedeutung dieser Zahl ist von Ihrem System abhängig. In POSIX, Windows und einigen anderen Systemen zählt diese Zahl die Anzahl der Sekunden von einer bestimmten Startzeit an (der "Epoche"). In anderen Systemen ist diese Bedeutung nicht spezifiziert. Die von der Zeit zurückgegebene Zahl kann nur als ein Argument für date und difftime verwendet werden.

os.tmpname ()

Gibt einen String mit einem Dateinamen zurück, der für eine temporäre Datei verwendet werden kann. Diese Datei muss vor ihrer Verwendung explizit geöffnet und bei Nichtverwendung explizit gelöscht werden. In einigen Systemen (POSIX) erstellt diese Funktion, zur Verhinderung von Sicherheitsrisiken, außerdem eine Datei mit einem Namen. (Zwischen dem Erhalt des Namens und der Erstellung der Datei könnte jemand anderes die Datei mit falschen Berechtigungen erstellen.) Zur Verwendung und zum Entfernen der Datei müssen Sie diese dennoch öffnen (selbst wenn Sie diese nicht verwenden).

Sie können, wenn möglich, auch io.tmpfile verwenden. Dadurch wird die Datei automatisch gelöscht, wenn das Programm endet.

10.19 Erweiterte Funktionsbibliothek

toboolan(value)

Wandelt den angegebenen Wert unter Verwendung der folgenden Regeln in einen booleschen Wert um: nil, boolean false, 0, empty string, „0“ String werden als false angesehen. Alles andere als true..

string.split(str, sep)

Teilt den angegebenen String durch das angegebene Trennzeichen in Stücke. Gibt eine LUA-Tabelle zurück.

knxlib.decodeia(indaddressa, indaddressb)

Konvertiert die binär-codierte individuelle Adresse in einen LUA-String. Diese Aktion akzeptiert entweder ein oder zwei Argumente (interpretiert als zwei einzelne Bytes).

knxlib.decodega(groupaddressa, groupaddressb)

Konvertiert die binär-codierte Gruppenadresse in einen LUA-String. Diese Aktion akzeptiert entweder ein oder zwei Argumente (interpretiert als zwei einzelne Bytes).

knxlib.encodega(groupaddress, separate)

Konvertiert den LUA-String in eine binär-codierte Gruppenadresse. Gibt Gruppenadressen eine einzelne LUA-Zahl zurück wenn das zweite Argument nil oder false ist. Gibt ansonsten zwei separate Bytes zurück.

ipairs (t)

Gibt drei Werte zurück: eine Iterator-Funktion, die Tabelle t, und 0, sodass der Aufbau die Paare (1,t[1]), (2,t[2]), ... bis zum ersten Ganzzahl-Schlüssel durchläuft, der nicht in der Tabelle vorhanden ist.

```
for i,v in ipairs(t) do body end
```

next (table [, index])

Erlaubt einem Programm das Durchqueren aller Felder der Tabelle. Das erste Argument ist eine Tabelle und das zweite Argument ist ein Index in dieser Tabelle. Next gibt den nächsten Index der Tabelle und dessen dazugehörigen Wert zurück. Wenn mit Nil als zweites Argument aufgerufen gibt next einen ersten Index und dessen dazugehörigen Wert zurück. Beim Aufruf mit dem letzten Index, oder mit Nil in einer leeren Tabelle, gibt next Nil zurück. Wenn das zweite Argument nicht vorhanden ist wird dieses als Nil interpretiert. Insbesondere können Sie next(t) zur Prüfung verwenden, ob die Tabelle leer ist. Die Reihenfolge, in der die Indizes aufgelistet werden, ist nicht spezifiziert, auch nicht für numerische Indizes. (Verwenden Sie zum Durchqueren einer Tabelle in numerischer Reihenfolge eine numerical for-, oder die ipairs-Funktion.) Das Verhalten von next ist undefiniert, wenn Sie während des Durchquerens einem nicht-existierenden Feld in der Tabelle einen beliebigen Wert zuweisen. Sie können jedoch die existierenden Felder modifizieren. Insbesondere können Sie vorhandene Felder löschen.

pairs (t)

Gibt folgende drei Werte zurück: Die Funktion next, die Tabelle t, und nil, sodass der Aufbau alle Schlüsselwert-Paare der Tabelle T durchläuft.

```
for k,v in pairs(t) do body end
```

tonumber (e [, base])

Versucht, sein Argument in eine Zahl umzuwandeln. Wenn das Argument bereits eine Zahl oder ein String ist, der in eine Zahl umgewandelt werden kann, gibt tonumber diese Zahl zurück. Ansonsten wird Nil zurückgegeben.

Ein optionales Argument spezifiziert die Basis zur Interpretation der Zahl. Die Basis kann eine beliebige Ganzzahl zwischen einschließlich 2 und 36 sein. In Basen über 10 repräsentiert der Buchstabe „A“ (entweder in Groß- oder Kleinbuchstaben) 10, „B“ repräsentiert 11, usw. „Z“ repräsentiert 35. Bei Basis 10 (dem Standard) kann die Zahl über einen Dezimal-Teil, als auch über einen optionalen Exponenten-Teil verfügen. In anderen Basen werden nur Ganzzahlen ohne Vorzeichen akzeptiert.

tostring (e)

Erhält ein Argument eines beliebigen Typs und wandelt diesen in einen String in angemessenem Format zurück. Zur kompletten Steuerung, wie Zahlen umgewandelt werden, verwenden Sie string.format.

Wenn der MetaTable on e über ein Feld „__tostring“ verfügt ruft tostring den betreffenden Wert mit e als

Argument auf und verwendet das Ergebnis des Aufrufs als sein Ergebnis.

type (v)

Gibt einen Typ seines einzelnen Arguments, codiert als String, zurück. Die möglichen Ergebnisse dieser Funktion sind „nil“ (ein String, nicht der Wert Nil), „number“, „string“, „boolean“, „table“, „function“, „thread“, und „userdata“.

11 Skript beispiele

11.1 Binär filter

Erstellen Sie zwei 1-Bit-Gruppenadressen unter der Registerkarte Objekt, wobei:

1/1/1 input

1/1/2 output

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es der Gruppe 1/1/1 hinzu. Das Skript wird jedes Mal ausgeführt, wenn die Gruppe 1/1/1 ein Telegramm empfängt.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
value_1 = grp.getvalue('1/1/1')
if value_1 == true then
-- do nothing
elseif value_1 == false then
grp.write('1/1/2', false)
end
```

11.2 Binär-Gate mit Bit-Gate

Erstellen Sie drei 1-Bit-Gruppenadressen unter der Registerkarte Objekt, wobei:

1/1/1 input

1/1/2 gate

1/1/3 output

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es der Gruppe 1/1/1 hinzu. Das Skript wird jedes Mal ausgeführt, wenn die Gruppe 1/1/1 ein Telegramm empfängt.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
value_1 = grp.getvalue('1/1/1') --input
value_2 = grp.getvalue('1/1/2') --gate
if value_2 == true then
-- do nothing
elseif value_2 == false then
grp.write('1/1/3', value_1) --output
end
```

11.3 Gate mit Byte-Gate

Erstellen Sie drei Gruppenadressen unter der Registerkarte Objekt, wobei:

1/1/1 input – beliebiger Typ, jedoch gleich wie output

1/1/2 gate- Byte-Object

1/1/3 output – gleich wie input

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es der Gruppe 1/1/1 hinzu. Das Skript wird jedes Mal ausgeführt, wenn die Gruppe 1/1/1 ein Telegramm empfängt.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
value_1 = grp.getvalue('1/1/1') -- input
value_2 = grp.getvalue('1/1/2') --gate
if value_2 == 0 then
-- do nothing
elseif value_2 < 0 or value_2 > 0 then
grp.write('1/1/3', value_1) --output
end
```

11.4 Or - Port (2 in 1 Out)

Erstellen Sie drei 1-Bit-Gruppenadressen unter der Registerkarte Objekt, wobei:

1/1/1 value 1

1/1/2 value 2

1/1/3 output

Fügen Sie den Gruppenadressen value1 und value2 den Tag OR1 hinzu.

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es Tag OR1 hinzu. Das Skript wird jedes Mal ausgeführt, wenn Gruppe 1/1/1 oder Gruppe 1/1/2 ein Telegramm empfangen.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
value_1 = grp.getvalue('1/1/1')
value_2 = grp.getvalue('1/1/2')
if value_1 == true or value_2 == true then
grp.write('1/1/3', true)
else
grp.write('1/1/3', false)
end
```

11.5 And - Port (2 in 1 Out)

Erstellen Sie drei 1-Bit-Gruppenadressen unter der Registerkarte Objekt, wobei:

1/1/1 value 1

1/1/2 value 2

1/1/3 output

Fügen Sie den Gruppenadressen value1 und value2 den Tag AND1 hinzu.

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es Tag AND1 hinzu. Das Skript wird jedes Mal ausgeführt, wenn Gruppe 1/1/1 oder Gruppe 1/1/2 ein Telegramm empfangen.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
value_1 = grp.getvalue('1/1/1')
value_2 = grp.getvalue('1/1/2')
if value_1 == true and value_2 == true then
    grp.write('1/1/3', true)
else
    grp.write('1/1/3', false)
end
```

11.6 Or - Port (5 in 2 Out)

Erstellen Sie Gruppenadressen unter der Registerkarte Objekt, wobei:

1/1/1 value 1 - 1bit

1/1/2 value 2 - 1bit

1/1/3 value 3 - 1bit

1/1/4 value 4 - 1bit

1/1/5 value 5 - 1bit

1/1/6 bit_output - 1bit

1/1/7 byte_output - 1byte

Fügen Sie den Gruppenadressen value1, value2, value3, value4 und value5 den Tag OR2 hinzu.

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es Tag OR2 hinzu. Das Skript wird jedes Mal ausgeführt, wenn die Gruppen 1/1/1, 1/1/2, 1/1/3, 1/1/4, 1/1/5 ein Telegramm empfangen.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
value_1 = grp.getvalue('1/1/1')
```

```
value_2 = grp.getvalue('1/1/2')
value_3 = grp.getvalue('1/1/3')
value_4 = grp.getvalue('1/1/4')
value_5 = grp.getvalue('1/1/5')
if value_1 == true or value_2 == true or value_3 == true or value_4 == true or value_5 == true then
grp.write('1/1/6', true) -- bit to 1
grp.write('1/1/7', 255) -- byte to 255
else
grp.write('1/1/6', false) -- bit to 0
grp.write('1/1/7', 0) -- byte to 0
end
```

11.7 And - Port (5 in 2 Out)

Erstellen Sie Gruppenadressen unter der Registerkarte Objekt, wobei:

1/1/1 value 1 - 1bit
1/1/2 value 2 - 1bit
1/1/3 value 3 - 1bit
1/1/4 value 4 - 1bit
1/1/5 value 5 - 1bit
1/1/6 bit_output - 1bit
1/1/7 byte_output - 1byte

Fügen Sie den Gruppenadressen value1, value2, value3, value4 und value5 den Tag AND2 hinzu.

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es Tag AND2 hinzu. Das Skript wird jedes Mal ausgeführt, wenn die Gruppen 1/1/1, 1/1/2, 1/1/3, 1/1/4, 1/1/5 ein Telegramm empfangen.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
value_1 = grp.getvalue('1/1/1')
value_2 = grp.getvalue('1/1/2')
value_3 = grp.getvalue('1/1/3')
value_4 = grp.getvalue('1/1/4')
value_5 = grp.getvalue('1/1/5')
if value_1 == true and value_2 == true and value_3 == true and value_4 == true and value_5 == true
then
grp.write('1/1/6', true) -- bit to 1
grp.write('1/1/7', 255) -- byte to 255
else
grp.write('1/1/6', false) -- bit to 0
grp.write('1/1/7', 0) -- byte to 0
end
```

11.8 Telegrammwandler (0/1 Bit in 0-255 Byte)

Erstellen Sie zwei Gruppenadressen unter der Registerkarte Objekt, wobei:

1/1/1 input – 1-bit
1/1/2 output – 1-byte

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es der Gruppe 1/1/1 hinzu. Das Skript wird jedes Mal ausgeführt, wenn die Gruppe 1/1/1 ein Telegramm empfängt.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
value_1 = grp.getvalue('1/1/1')
if value_1 == true then -- bit value (in)
grp.write('1/1/2', 255) -- byte value (out)
else
grp.write('1/1/2', 0) -- byte value (out)
end
```

11.9 Wert vergleichen

```
value_1 = grp.getvalue('1/1/1')
value_2 = grp.getvalue('1/1/2')
if value_1 == value_2 then
grp.write('1/1/3', true) -- bit to 1
grp.write('1/1/4', 255) -- byte to 255
else
grp.write('1/1/3', false) -- bit to 0
grp.write('1/1/4', 0) -- byte to 0
end
```

11.10 Speicherbildschirm 1 (RGB Wert)

```
value_1 = grp.getvalue('1/1/1') --RED
value_2 = grp.getvalue('1/1/2') --GREEN
value_3 = grp.getvalue('1/1/3') --BLUE
storage.set('Scene1_Red', value_1)
storage.set('Scene1_Green', value_2)
storage.set('Scene1_Blue', value_3)
```

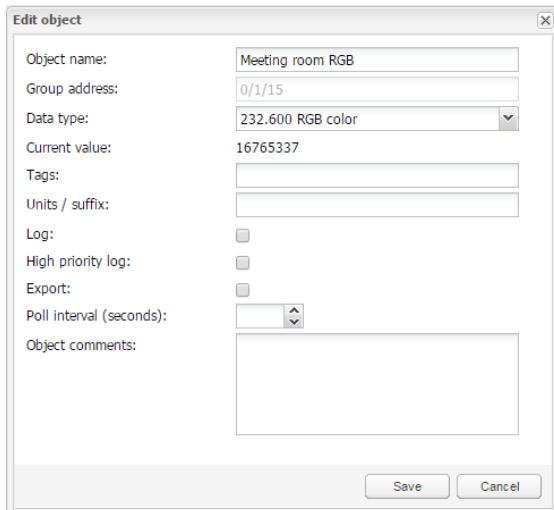
11.11 Aufrufbildschirm 1 (RGB Wert)

```
value_1 = storage.get('Scene1_Red')
value_2 = storage.get('Scene1_Green')
value_3 = storage.get('Scene1_Blue')
if not value_1 then
--if storage value does not exist do nothing
else
grp.write('1/1/1', value_1) --RED
end
if not value_2 then
--if storage value does not exist do nothing
else
grp.write('1/1/2', value_2) --GREEN
end
if not value_3 then
--if storage value does not exist do nothing
else
grp.write('1/1/3', value_3) --BLUE
end
```

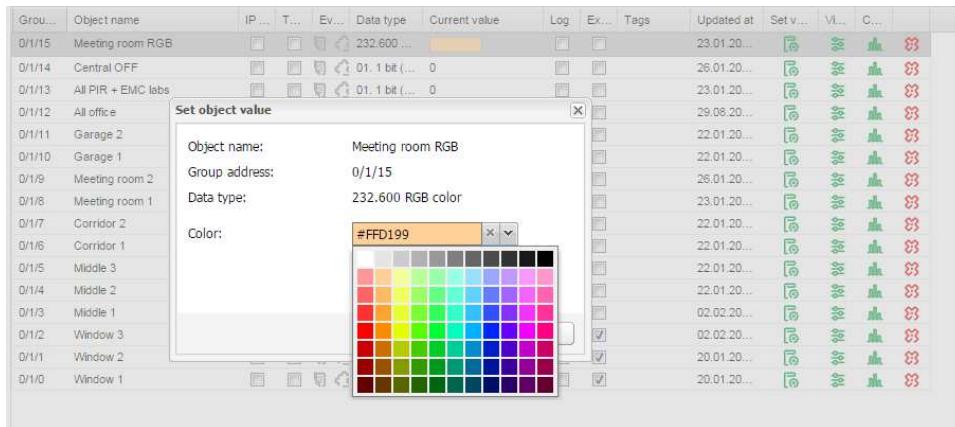
11.12 RGB-Objekt

11.12.1 Arbeiten mit dem RGB-Objekt

- Erstellen Sie ein Objekt und definieren Sie die Objekt-Parameter in Konfigurator/Objekte/doppelklicken Sie auf das Objekt

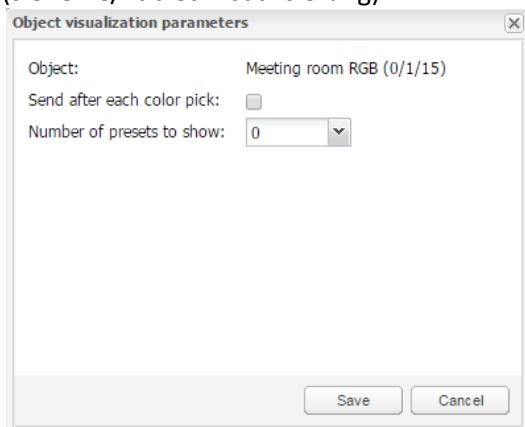


- Legen Sie die RGB-Farbe in *Konfigurator/Objekte/Wert festlegen, fest*

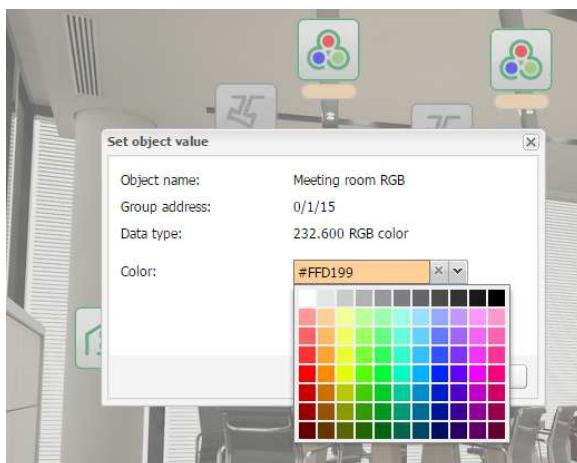


- Legen Sie die Parameter zur Objektvisualisierung in *Konfigurator/Objekte/Vis. fest Params*

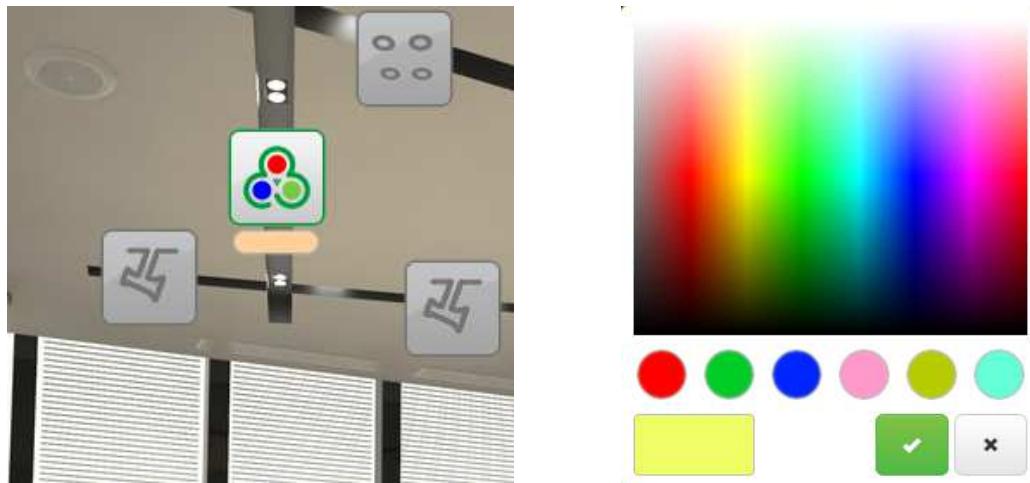
Wenn die Option Nach jeder Farbaufnahme senden markiert ist, wird nach der Freigabe der linken Maustaste (PC), oder nach dem Entfernen des Fingers (Touch-Screen) automatisch ein neu aktualisiertes Objekt mit der ausgewählten Farbe an den Bus zum Visualisierung-Bildschirm gesendet.
(siehe PC/Tablet-Visualisierung)



- Legen Sie die RGB-Farbe in Konfigurator/Visualisierung fest (identisch mit Konfigurator/Objekte/Wert festlegen)



- Legen Sie die Farbe als Benutzer in PC/Tablet-Visualisierung fest



11.12.2 Skript zur Steuerung der RGB LED-Farbe

-- Dieses Skript teilt 1 x 3 Byte RGB-Objekte in 3 x 1 Byte Wertobjekte
-- Erstellen Sie ein ereignisbasiertes Skript, welches vom 3 Byte RGB-Objekt ausgeführt wird
-- Füllen Sie die konfigurierbaren Parameter mit Gruppenadressen oder Gruppennamen
-- RGB und 1 Byte-Objekte müssen vom korrekten Datentyp sein

----- Konfigurierbare Parameter -----

```
redGroup = '1/1/5' --- modifizieren entweder Gruppenadresse oder Name der Gruppe  
greenGroup = 'LED1 Green Value' --- modifizieren entweder Gruppenadresse oder Name der Gruppe  
blueGroup = '1/1/7' --- modifizieren entweder Gruppenadresse oder Name der Gruppe
```

```
value = event.getvalue()  
  
Blue= bit.band(value, 0xFF)  
Green = bit.rshift(bit.band(value, 0xFF00), 8)  
Red = bit.rshift(bit.band(value, 0xFF0000), 16)  
  
grp.write(redGroup, Red, dt.uint8)  
grp.write(greenGroup, Green, dt.uint8)  
grp.write(blueGroup, Blue, dt.uint8)
```

Dieses Skript teilt ein 3 Byte RGB-Objekt in drei 1 Byte Wertobjekte, das die einzelnen Farben Rot/Grün/Blau steuert.#

11.12.3 Skript zum Auslösestatus vom RGB LED

-- Dieses Skript konvertiert 3 x 1 Byte Statusobjekte in 1 x 3 Byte RGB-Objekt
-- Erstellen Sie ein ereignisbasiertes Skript, welches von einem eindeutigen TAG ausgeführt wird.
Dieser Tag
-- muss den Rot, Grün, Blau 1 Byte Statusobjekten angehängt werden

-- Füllen Sie die konfigurierbaren Parameter mit Gruppenadressen oder Gruppennamen
-- Für Eingaben wird die Verwendung eines 1 Byte Statusobjekts empfohlen
-- RGB-Objekt muss vom korrekten Datentyp sein

----- Konfigurierbare Parameter -----

```
redGroup = '1/1/1' --- modifizieren entweder Gruppenadresse oder Name der Gruppe  
greenGroup = 'LED1 Green Status' --- modifizieren entweder Gruppenadresse oder Name der Gruppe  
blueGroup = '1/1/3' --- modifizieren entweder Gruppenadresse oder Name der Gruppe
```

```
rgbGroup = 'RGB Value' --- modifizieren entweder Gruppenadresse oder Name der Gruppe
```

```
red = grp.find(redGroup)  
green = grp.find(greenGroup)  
blue = grp.find(blueGroup)
```

```
redHex = red.datahex  
greenHex = green.datahex  
blueHex = blue.datahex
```

```
RGB = lmcore.hextoint(redHex..greenHex..blueHex)
```

```
grp.write(rgbGroup, RGB)
```

11.13 Hysterese

(Objekt 1/1/2 nicht ändern, wenn der Wert der Objekte 1/1/1 zwischen 100 und 200 liegt)

```
value_1 = grp.getvalue('1/1/1') -- Byte-Wert  
if value_1 < 100 then  
    grp.write('1/1/2', false) -- bit to 0  
  
elseif value_1 > 200 then  
    grp.write('1/1/2', true) -- bit to 0  
end
```

11.14 Zufälliger Byte-Wert

```
steps = 255 -- mögliche Schritte ändern diesen Wert zur Verringerung des Werts, um größere Schritte  
durchzuführen  
random = math.random(0, (steps - 1)) * 255 / (steps - 1)  
outcome = (math.floor(random))  
value_1 = grp.getvalue('1/1/1')  
grp.write('1/1/1', outcome) -- Zufälligen Byte-Wert auf Objekt schreiben
```

11.15 Zyklischer Repeater (Verzögerung 60 Sekunden)

```
value_1 = grp.getvalue('1/1/1')
if value_1 == true then
repeat
value_1 = grp.getvalue('1/1/1')
if value_1 == true then
grp.write('1/1/2', true)
-- für 60 Sekunden warten
os.sleep(60)
end
until value_1 == false
end
```

11.16 Stepper / Zähler Positive Eingabe

```
value_1 = grp.getvalue('1/1/1') -- Positive Eingabe
if value_1 == true then
Stepper_Value = storage.get('Value_Stepper_1')
if not Stepper_Value then
Stepper_Value = 0
end
if Stepper_Value == 255 then
else
Stepper_Value = Stepper_Value + 1
end
storage.set('Value_Stepper_1', Stepper_Value)
grp.write('1/1/4', Stepper_Value)
end
```

11.17 Stepper / Zähler Negative Eingabe

```
value_1 = grp.getvalue('1/1/2') -- Negative input
if value_1 == true then
Stepper_Value = storage.get('Value_Stepper_1')
if not Stepper_Value then
Stepper_Value = 0
end
if Stepper_Value == 0 then
else
Stepper_Value = Stepper_Value - 1
end
storage.set('Value_Stepper_1', Stepper_Value)
grp.write('1/1/4', Stepper_Value)
end
```

11.18 Stepper / Zähler zurücksetzen

```
value_1 = grp.getvalue('1/1/3')
if value_1 == true then
storage.set('Value_Stepper_1', 0)
grp.write('1/1/4', 0)
end
```

11.19 Anschaltverzögerung (Schaltfläche auf „nur intern aktualisieren“ setzen)

```
value_1 = grp.getvalue('1/1/1')
if value_1 == true then
os.sleep(3) -- Verzögerungszeit
grp.write('1/1/1', true)
end
```

11.20 Durchschnitt

```
value_1 = grp.getvalue('1/1/1')
value_2 = grp.getvalue('1/1/2')
Average = value_1 + value_2
Average = (Average / 2)
value_3 = grp.getvalue('1/1/3')
grp.write('1/1/3', true)
```

11.21 Abschaltverzögerung

```
value_1 = grp.getvalue('1/1/1')
if value_1 == true then
os.sleep(3) -- Verzögerungszeit
grp.write('1/1/1', false)
end
```

11.22 Timer Treppenaufgang (mit variablem Zeitobjekt)

```
value_1 = grp.getvalue('1/1/1')
value_2 = grp.getvalue('1/1/2') -- Wert der Variablen
if value_1 == true then
os.sleep(value_2)
grp.write('1/1/1', false)
end
```

11.23 Wertspeicher (in Speicher schreiben)

```
value_1 = grp.getvalue('1/1/1')
storage.set('Storage_Value_Memory_1', value_1)
```

11.24 Wertspeicher (von Speicher holen)

```
Value_Memory_1 = storage.get('Storage_Value_Memory_1')
if not Value_Memory_1 then
-- do nothing
else
grp.write('1/1/1', Value_Memory_1)
end
```

11.25 Multiplexer (1 in / 3 out)

HINWEIS: Der Objekttyp muss der gleiche sein.

```
value_1 = grp.getvalue('1/1/1')
grp.write('1/1/2', Value_1)
grp.write('1/1/3', Value_1)
grp.write('1/1/4', Value_1)
```

11.26 Rundungsfunktion mit Verwendung allgemeiner Funktionen

Fügen Sie den allgemeinen Funktionen folgenden Code hinzu

```
-- Rundet eine Zahl auf die angegebene Anzahl der Dezimalstellen...
function round(num, idp)
local mult = 10^(idp or 0)
return math.floor(num * mult + 0.5) / mult
end
```

Skript im Skript-Editor erstellen

```
-- Round-Funktion (mit globaler Funktion)
value_1 = grp.getvalue('1/1/1')
round(value_1, 2) -- mit Verwendung der Rundungsfunktion von allgemeinen Funktionen
grp.write('1/1/1', Value_2)
```

11.27 Daten und Zeit in KNX-Gruppenadressen schreiben

```
-- aktuelle Daten als Tabelle erhalten
now = os.date('*t')
-- Wochentag des Systems starten mit Sonntag, in KNX-Format konvertieren
wday = now.wday == 1 and 7 or now.wday - 1
-- Zeittabelle
time = {
    day = wday,
    hour = now.hour,
    minute = now.min,
    second = now.sec,
}
-- Datumstabelle
date = {
    day = now.day,
    month = now.month,
    year = now.year,
}
-- auf Bus schreiben
grp.write('1/1/2', time, dt.time)
grp.write('1/1/1', date, dt.date)
```

11.28 Daten auf Gruppen mit Tags schreiben

Erstellen Sie einige 1-Bit-Gruppenadressen und fügen Sie diesen den Tag „Light“ hinzu
Erstellen Sie eine oder mehrere Gruppen, die sich von der anderen unterscheiden, um das Skript auszulösen.

1/1/1 – Light1 – Tag „Light“
1/1/2 – Light2 – Tag „Light“
1/1/3 – Light3 – Tag „Light“
1/1/4 – Light4 – Tag „Light“
1/1/5 – Light5 – Tag „Light“
1/1/6 – Light6 – Tag „Light“

1/1/10 – Szene aktive Gruppe – **kein Tag hinzugefügt!**

Erstellen Sie ein ereignisbasiertes Skript und fügen Sie es der Gruppe 1/1/10 hinzu. Das Skript wird jedes Mal ausgeführt, wenn die Gruppe 1/1/10 ein Telegramm empfängt.

Fügen Sie den Skript-Editor den folgenden Code hinzu:

```
AllLights = grp.tag('Light')
AllLights: write(true)
```

Es werden jedes Mal, wenn die Gruppe 1/1/10 ein Telegramm empfängt, alle Lichter angeschaltet.

HINWEIS:Starten Sie das Skript nicht vom gleichen Tag oder einer Gruppenadresse, die den gleichen Tag enthält. Dies erzeugt eine Endlosschleife, die Traffic am Bus erzeugt und zu einer hohen Last am Prozessor führt.

Sollte eine Endlosschleife erzeugt worden sein, so stoppen Sie das Skript und starten Sie spaceLYnk neu.

12 Anwendungshinweise

Die Dokumente mit Anwendungshinweisen geben eine detaillierte Beschreibung des ausgewählten Themas mit Fokus auf die einfache Einbindung. Die Liste der Dokumente mit Anwendungshinweisen wird kontinuierlich aktualisiert.

12.1.1 Liste mit aktuell herausgegebenen Anwendungshinweisen:

- AN001_spaceLYnk-Integration mit Verwendung von BACnet_v1.4
- AN003_Modbus-Leistungsmesser und spaceLYnk_v1.3
- AN004_spaceLYnk und SmartStruxureLite-Integration via Web-Services
- AN005_PIR-Trend und Protokolle mit spaceLYnk
- AN006_Erweiterte Techniken in der Visualisierung (spaceLYnk)_v1.2
- AN008_spaceLYnk und SmartStruxureLite-Integration via Web-Services
- AN009_spaceLYnk und SmartStruxureLite-Integration via Modbus TCP
- AN010_RS232-Steuerung mit spaceLYnk
- AN011_E-Mail, SMS und FTP in spaceLYnk
- AN013_Wetterbericht in spaceLYnk abrufen
- AN015/Desktop-Shortcut zur Visualisierung von spaceLYnk
- AN016_spaceLYnk als Modbus-Slave
- AN017_Adressierbare Einschränkungen des spaceLYnk-Controller_v1.1
- AN023_Redundanter Betriebsmodus von zwei spaceLYnks
- AN024_Block-Funktionen für spaceLYnk- und homeLYnk-Controller

12.1.2 Verfügbarkeit der Anwendungshinweise

Alle Dokumente mit Anwendungshinweisen können auf der Website von Schneider Electric heruntergeladen werden.

Zum besseren Auffinden des Dokuments kann das Suchfeld oben an der Website von Schneider Electric verwendet werden.