

SENSORhub SEH

Revision A | 2026-07-23

Übersicht

Der SENSORhub SEH ist ein Controller ("Hub"), der adressierbare LED-Streifen und angeschlossene SMC-Taster (SENSORswitches) für Führungs- und Pick-by-Light-Anwendungen ansteuert.

Der Hub unterstützt bis zu 5 LED-Streifen (LED_STRIP_1 bis LED_STRIP_5). Jeder Streifen kann mehrere Segmente enthalten, und jedes Segment kann individuell mit einer Start-/Stopp-Position, einem Effekt, einer Geschwindigkeit und bis zu drei Farben konfiguriert werden.

Steuerungsschnittstellen

Der SENSORhub SEH kann über zwei parallel arbeitende Schnittstellen gesteuert und überwacht werden — der zuletzt empfangene Befehl bestimmt den aktuellen Zustand:

- **MQTT Communication** — Geräte-Onboarding, Konfiguration und Aktivierung der LED-Streifen, Status-/Health-Reporting, Firmware-Updates und Zeitverteilung.
- **Modbus Interface** — Modbus-TCP-Steuerung der LED-Streifen und SMC-Taster sowie schreibgeschützte Statusregister.

Geräte-Varianten

Gerät	Beschreibung
SEH11	SENSORhub-Controller nur für SMC-Taster (SENSORswitches) — keine LED-Streifen-Unterstützung
SEH101	SENSORhub-Controller für bis zu 5 adressierbare LED-Streifen, mit Ausgangsdiagnose (Verdrahtungsfehler-Erkennung an den Ausgangsklemmen)
SEH201	SENSORhub-Controller für bis zu 5 adressierbare LED-Streifen und SMC-Taster (SENSORswitches), mit Ausgangsdiagnose

HINWEIS

Einige Funktionen sind geräteabhängig. LED-Streifen (bis zu 5) werden von SEH101 und SEH201 unterstützt; SMC-Taster werden von SEH11 und SEH201 unterstützt. Die Ausgangsdiagnose ist bei SEH101 und SEH201 verfügbar; Diagnosemeldungen werden bei SEHx01-Geräten gemeldet. Details dazu finden Sie auf den jeweiligen Schnittstellenseiten.

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

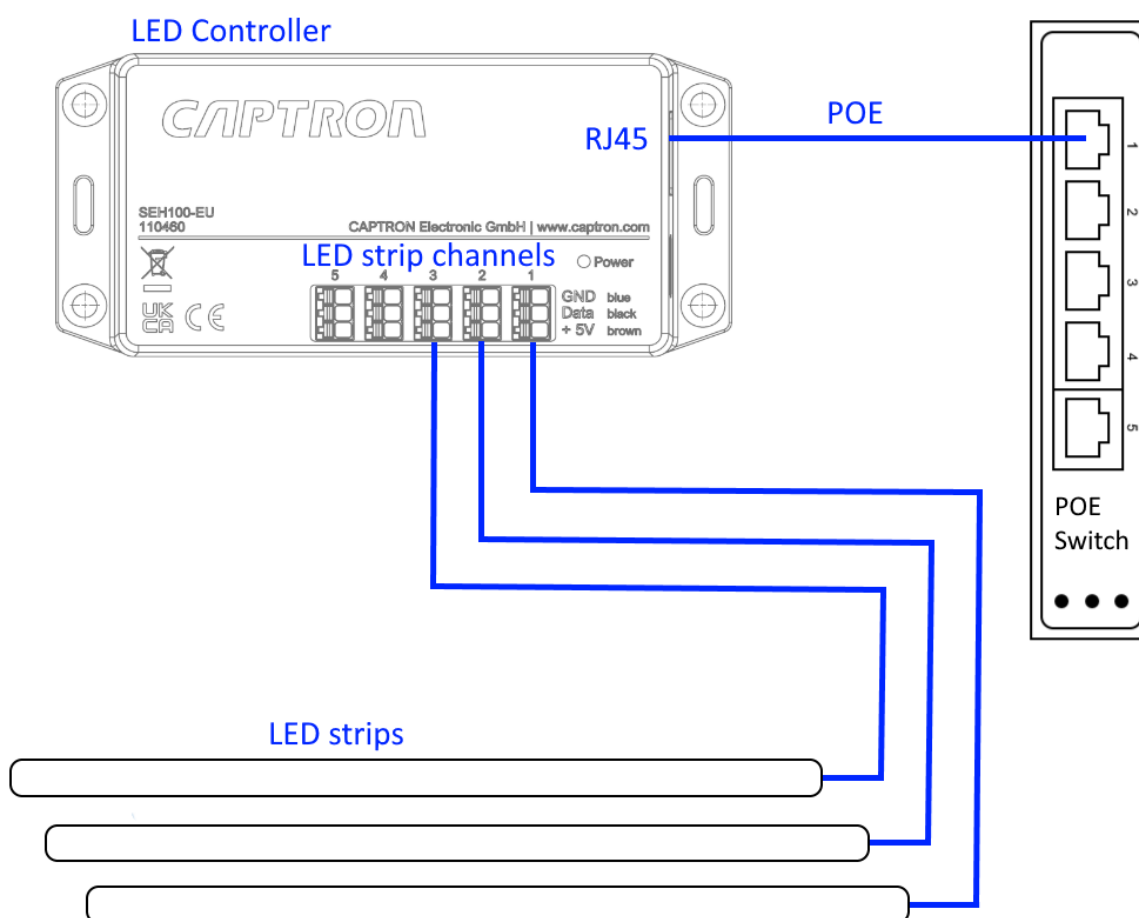
Ablehnen

Akzeptieren

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Hardware-Montage

Die Kabelverbindungen zwischen allen Teilen des Systems sind im folgenden Schema dargestellt.



Jeder LED-Streifen wird an einen der LED-Streifen-Kanäle (Ausgänge 1-5) angeschlossen. Die Adern des Streifens sind: **GND** (blau), **Data** (schwarz) und **+5V** (braun).

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Ablehnen

Akzeptieren

Konnektivität

Diese Seite beschreibt, wie sich ein SENSORhub SEH Gerät mit dem Netzwerk verbindet, wie sein Verbindungszustand signalisiert wird, wie es offline über die Web-Setup-Seite konfiguriert wird und wie es zurückfällt (Fallback), wenn die konfigurierte Verbindung nicht hergestellt werden kann.

Statussignalisierung

Gilt für SEH101 und SEH201.

Beim Start wird der Gerätestatus durch die angeschlossenen LED-Streifen und die Status-LED (SEHx01) signalisiert. Zur Unterstützung während der Installation (Zuordnung der Streifen zu den Ausgängen) leuchtet an jedem Ausgang eine unterschiedliche Anzahl von LEDs auf: eine LED an Ausgang 1, zwei LEDs an Ausgang 2 und so weiter.

Die Signalisierungsfarben werden angezeigt, bis der erste Set-/Data-Befehl empfangen wird.

Farbe	Status
Rot	Ethernet-/WiFi-Verbindungsversuch, keine Ethernet-/WiFi-Verbindung.
Gelb	MQTT-Verbindungsversuch, keine MQTT-Verbindung.
Blau	Fallback-Mechanismus aktiv (geplant ab V0.2.33/V0.3.6), Konfiguration und/oder Netzwerkverbindung prüfen.
Cyan/Grün-Blau	Verbindung hergestellt — zum Onboarding-Service (wenn kein Benutzer-MQTT konfiguriert ist) oder zum Benutzer-MQTT (wenn konfiguriert).

Web-Setup-Seite

Gilt für SEH101 und SEH201.

Als Möglichkeit zur Offline-Konfiguration kann die integrierte Setup-Seite verwendet werden. Sie ist über jeden Browser erreichbar, indem Sie `http://` gefolgt von der IP-Adresse des Geräts eingeben. Standardmäßig versucht das Gerät, seine IP-Adresse von einem DHCP-Server zu beziehen; die zugewiesene Adresse kann in diesem Server/Router nachgeschlagen werden. Die Web-Setup-Seite ist ab Firmware-Version V0.2.27-29 verfügbar.



Unique ID:	mle_python_revB
MQTT Broker:	192.168.178.93
MQTT Port:	1883
MQTT User:	null
MQTT Password:	
MQTT TLS:	☐
Power Limit:	8.000000
Wifi SSID:	null
Wifi Passphrase:	null
DHCP:	☒
IP address:	null
Subnet mask:	null
Gateway:	null
DNS:	null
Password:	
apply	

For unused entries (e.g. mqtt authentication, Wifi) set the field content "null".

Um die Einstellungen zu übernehmen, geben Sie `pythonhead22` in das Passwortfeld (letzte Zeile) ein und bestätigen Sie mit der Schaltfläche „apply“.

HINWEIS

Setzen Sie den Feldinhalt für nicht verwendete Einträge (zum Beispiel MQTT-Authentifizierung oder Wifi) auf `null`.

Statische IP-Einstellung

Wenn eine statische IP-Einstellung erforderlich ist (zum Beispiel, wenn kein DHCP-Server verwendet werden kann), füllen Sie mindestens die drei Felder aus: IP-Adresse, Subnetzmaske und Gateway. Wenn kein Gateway vorhanden ist, geben Sie hier dennoch eine gültige Adresse ein (zum Beispiel die eigene Adresse des Geräts).

Ab Firmware-Version V0.2.27-57 kann das DNS-Feld auf `null` belassen werden. Das Gerät verzichtet dann darauf, sich mit dem Onboarding-Service zu verbinden.

Zertifikate für die Benutzer-MQTT-Verbindung

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer

Wenn TLS zusammen mit Client-Zertifikaten (mTLS) verwendet werden soll, bietet die Web-Schutzseite auch die Möglichkeit, Zertifikatsdateien hochzuladen (zum Beispiel `crt` / `pem` / `key`).

Dies erfordert mindestens Firmware-Version V0.3.7. Dateiname und Dateiendung sind ohne Bedeutung. Stellen Sie sicher, dass das TLS-Kontrollkästchen aktiviert ist.

Stellen Sie außerdem sicher, dass Sie das passende Aussteller-Zertifikat als CA-Zertifikat hochladen, damit die TLS-Engine sowohl das Gerätezertifikat als auch das eingehende Serverzertifikat verifizieren kann. Es können auch selbstsignierte Zertifikate verwendet werden.

Wenn die Datei erfolgreich hochgeladen wurde, erscheint „file loaded“. Um eine Datei zu überschreiben, wählen Sie eine neue Datei aus und klicken Sie auf „Upload“. Alle Dateien können mit „Remove“ entfernt werden. Nachdem Sie alle erforderlichen Dateien hochgeladen haben, starten Sie das Gerät durch Klicken auf „Restart“ neu.

Fallback-Funktionalität

Gilt für SEH101 und SEH201.

Priorität

Ab Firmware-Version V0.2.27-41 hat der Ethernet-Anschluss Vorrang vor WiFi. Wenn über LAN eine Verbindung hergestellt werden kann, wird die WiFi-Einstellung ignoriert. Wenn keine Ethernet-Verbindung hergestellt werden kann und kein WiFi konfiguriert ist, versucht das Gerät, sich mit dem Fallback-WiFi zu verbinden (nur EU-Varianten) — dies dauert bis zu 45 Sekunden.

WiFi-Fallback

Wenn ein konfiguriertes WiFi-Netzwerk nicht erreicht werden kann, versucht das Gerät, auf dieses fest hinterlegte WiFi zuzugreifen:

- SSID `SEH_fallback_wifi`
- Passphrase `fallback_pw`

Dieses Fallback-WiFi-Netzwerk kann zum Beispiel durch Einrichten eines Hotspots auf einem Mobiltelefon mit dem entsprechenden Namen und der entsprechenden Passphrase erstellt werden.

Sobald die Verbindung zum Fallback-WiFi hergestellt ist, verbindet sich das Gerät mit dem Onboarding-Service und erhält die aktuelle Konfiguration. Auf diese Weise kann eine fehlerhafte WiFi-Konfiguration korrigiert werden.

HINWEIS

Die WiFi-Funktionalität ist nur bei EU-Varianten verfügbar. Bei anderen Varianten wie JP, CN oder US ist das WiFi-Modul deaktiviert.

Fallback bei statischer IP

Wenn ein Gerät mit statischer IP-Einstellung nach etwa 45 Sekunden nicht auf das Netzwerk zugreifen kann, führt es einen Fallback durch und versucht, seine Netzwerkeinstellungen von einem DHCP-Server zu beziehen.

Wenn der Fallback erfolgreich ist, verbindet sich das Gerät mit dem Onboarding-Service und erhält die aktuelle Konfiguration. Eine fehlerhafte IP-Einstellung kann auf diese Weise korrigiert werden.

DHCP-Fallback (APIPA)

Wenn das Gerät so konfiguriert ist, dass es eine Adresse von einem DHCP-Server bezieht, und dieser Server nicht erreichbar ist, weist sich das Gerät nach etwa 45 Sekunden diese Netzwerkeinstellung (innerhalb des APIPA-Bereichs) selbst zu:

- IP-Adresse: 169.254.101.201
- Subnetz: 255.255.0.0

Mit einer direkten Ethernet-Verbindung zu einem Computer kann es über die Web-Setup-Seite neu konfiguriert werden. Diese Fallback-Methode ist ab Firmware-Version V0.2.27-63 verfügbar.

Dreistufiger Fallback

Ab Firmware-Version V0.3.7-1 sind die Fallback-Mechanismen vereinfacht und vereinheitlicht.

Unabhängig von der Konfiguration (statische IP oder DHCP) sind die Fallback-Schritte immer:

Statische IP → DHCP → APIPA → dann wieder von vorne beginnen. Jeder Schritt ist etwa 45 Sekunden lang aktiv.

MQTT Communication

SENSORhub SEH-Geräte werden über MQTT gesteuert und überwacht. Diese Seite beschreibt die MQTT-Topics und deren Nachrichten-Payloads. Parallel dazu steht eine Modbus-TCP-Schnittstelle zur Verfügung — siehe [Modbus Interface](#).

Definitionen

- `{product}` ist der Name des Produkts, z. B. `SEH101` oder `SEH201`.
- `{device-id}` ist der eindeutige Name des Geräts und besteht aus einer zufällig generierten Wortkombination wie `GroovySquareTongue` oder `TinyHappyGroup`. Diese ID wird zum Aufbau der Topics verwendet und ist auf dem Gerät aufgedruckt.
- `{Content definition}` ist der Teil des Topics nach der `{device-id}`, z. B. `/Pub/MAM`.

Alle Topics folgen diesem Muster:

```
captron.com/{product}/nd/{device-id}/{Content definition}
```

Geräteinformationen

Publish — `/Pub/MAM`

Richtung: Gerät publiziert.

```
{
  "Content": "/Pub/MAM",
  "BoardName": "lucky_python",
  "Manufacturer": "CAPTRON",
  "Model": "PYTHON-Head",
  "ProductCode": "123456789",
  "SoftwareVersion": "v0.0.1"
}
```

Das Gerät sendet Daten, die generisch und geräteindividuell sind, die Konfiguration, die dem Gerät während des Onboarding-Prozesses übergeben wurde, sowie alle Methoden, die aufgerufen werden können.

Subscribe — `/Get/MAM`

Richtung: Gerät abonniert. Payload: keine.

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Geräteinformationen anfordern. Das Gerät antwortet auf dem entsprechenden Topic

Ablehnen

Akzeptieren

Konfiguration

Subscribe — /Set/Config/MQTT

Richtung: Gerät abonniert.

```
{
  "Content": "/Set/Config/MQTT",
  "Vendor": "captron",
  "Port": 1883,
  "MQTTServer": "update.oneGrid.captron.com",
  "MQTTUsername": "Captron2022",
  "MQTTPassword": "qwxYT321!",
  "Encryption": false
}
```

Übermittelt die Verbindungsinformationen, die spezifisch für den Betreiber der Anwendung sind, wie z. B. die Serverkonfiguration. Werden keine Anmeldedaten benötigt, ist `null` als Wert für `MQTTUsername` und `MQTTPassword` zu verwenden.

Diese Nachricht wird während des automatischen Onboarding-Prozesses verwendet. Beachten Sie, dass das Gerät nach einem Stromzyklus diese Informationen erneut vom Onboarding-System anfordert.

Subscribe — /Set/Config/LedStrip

Richtung: Gerät abonniert.

```
{
  "Content": "/Set/Config/LedStrip",
  "Demo": false,
  "LED_STRIP_1": {
    "ColorAlignment": {"Active": true, "R": 255, "G": 180, "B": 255}
  },
  "LED_STRIP_2": {
    "ColorAlignment": {"Active": true, "R": 255, "G": 180, "B": 255}
  },
  etc...
}
```

Anwendungsfallspezifische Konfiguration der LED-Streifen (z. B. Länge als Anzahl der LEDs).

- Ab FW v0.2.10 ist diese Nachricht nicht mehr zwingend erforderlich. Standardmäßig werden 180 LEDs pro Streifen verwendet.
- Der boolesche Wert `Demo` aktiviert/deaktiviert den Demo-Modus. Ist dieser aktiviert, spielt das Gerät unterschiedliche Farben und Effekte auf Ausgang 1 und 2 ab. Diese Einstellung wird dauerhaft im Speicher abgelegt – wird der Demo-Modus aktiviert, bleibt er auch nach einem Stromzyklus aktiv.

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

- Ab FW V0.3.10-13-G225F748 ist die Farbausrichtung für jeden LED-Streifen möglich. Setzen Sie das Flag `Active` auf `true` und stellen Sie die entsprechenden RGB-Werte im Bereich 0-255 ein.

LED-Streifen aktivieren

Subscribe — `/Set/Data/LedStrip`

Richtung: Gerät abonniert.

```
{
  "Content": "/Set/Data/LedStrip",
  "LED_STRIP_1": {
    "Active": true,
    "Segments": [
      {
        "StartLED": 0,
        "StopLED": 30,
        "Speed": 190,
        "Effect": 1,
        "Colors": [
          {
            "R": 0,
            "G": 150,
            "B": 0
          },
          {
            "R": 0,
            "G": 150,
            "B": 0
          }
        ]
      }
    ]
  }
}
```

Schaltet die LEDs ein.

- Verwenden Sie das Flag `Active`, um den Streifen ein- oder auszuschalten.
- Effekte:
 - 1: FX_MODE_STATIC
 - 2: FX_MODE_BLINK
 - 3: FX_MODE_FLASH
 - 4: FX_MODE_LEADING_LINES
 - 5: FX_MODE_LEADING_LINES_REVERSE
 - 6: FX_MODE_MULTIPICKER
 - 7: FX_MODE_PICK (benötigt mind. FW-Version V0.3.7)
 - 8: FX_MODE_PLACE (benötigt mind. FW-Version V0.3.7)

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

- 200: FX_MODE_CUSTOM_KNIGHT_RIDER
- Geschwindigkeit von 1 (langsam) bis 250 (schnell).
- Mehrere Farben gelten nur für bestimmte Effekte wie FX_MODE_MULTIPICKER. In diesem Fall wechseln die Farben zwischen den im Array `Colors` angegebenen RGB-Werten. Bei allen anderen Effekten gilt nur die erste Farbe im Array.
- Es können 6 Segmente mit unterschiedlichen Farben/Effekten gesetzt werden. Je nach maximaler MQTT-Nachrichtengröße können weitere Segmente gesetzt werden. Die MQTT-Nachrichtengröße muss unter 4k liegen.
- Dieser Befehl kann ignoriert werden, wenn eine Leistungsgrenze erreicht wird. In diesem Fall wird eine Fehlermeldung protokolliert, um den Benutzer zu informieren.

Steuerung der SMC-Taster (Geräte SEH20x/SEH11)

Die Steuerung der SMC-Taster (SENSORswitches) ist bei SEH11 und SEH201 verfügbar. Die an den Hub angeschlossenen SMC-Taster werden über ein einziges Topic gesteuert. Die Modbus-Schnittstelle bietet dieselbe SMC-Steuerung parallel an — siehe [Modbus Interface](#).

Implementiert in FW V0.2.5.

Subscribe — `/Set/Data/Smc`

Richtung: Gerät abonniert.

```
{
  "Content": "/Set/Data/Smc",
  "Address": "{Button address, HUB or ALL_SENSORS}",
  "CommandType": "{Command Type}",
  "Offset": "{Offset}",
  "Payload": "{Payload}"
}
```

- **Address** — eine bestimmte Taster-Adresse, `HUB` (der SEH selbst) oder `ALL_SENSORS` (Broadcast an alle Taster).
- **CommandType** und **Offset** bestimmen gemeinsam, welcher Befehl ausgeführt wird und wie die Zeichenkette `Payload` interpretiert wird.

Folgende Befehle stehen zur Verfügung:

Befehl	CommandType	Offset	Payload
LED-Ring und Anzeige setzen	SET_PARAMETER	PRE_BUTTON_MODE oder POST_BUTTON_MODE	{ENABLED, DISABLED or LONGPRESS ENABLED}/{Quantity}/{LED ring color}/{LED ring effect}/{Display text}
Tasterausrichtung setzen	SET_PARAMETER	BUTTON_ROTATE	BUTTON_ROTATION_NORMAL oder BUTTON_ROTATION_UPSIDEDOWN
Longpress-Zeit setzen	SET_PARAMETER	LONGPRESS_TIME	Millisekunden geteilt durch 10 (max. 2,55 Sekunden)
Taster-Adresse zuweisen	SET_STATUS	ADDRESS_TO_BE_ASSIGNED	Zuzuweisende Nummer
Spezialbefehle (Pre-/Post-Zustand umschalten, Neustart usw.)	SET_STATUS	SPECIAL_COMMANDS	PRE_PRESSED_STATE, POST_PRESSED_STATE, REBOOT, SHOW_CURRENT_ADDRESS oder LONG_PRESSED_STATE
Polling-Funktion (Address = HUB)	SET_PARAMETER	(keiner)	Polling-Adressen getrennt durch / (bis zu 6), z. B. 1002/1003/1004; 0000 schaltet das Polling aus

Spezialbefehle (Offset SPECIAL_COMMANDS):

- PRE_PRESSED_STATE — in den Pre-Zustand setzen
- POST_PRESSED_STATE — in den Post-Zustand setzen
- REBOOT — Taster neu starten
- SHOW_CURRENT_ADDRESS — Taster-Adresse anzeigen
- LONG_PRESSED_STATE — in den Longpress-Zustand setzen (benötigt mind. SEH-Firmware V0.2.27-3)

HINWEIS

Ältere Firmware-Versionen übernehmen die Einstellung der **Tasterausrichtung** nach einem Power-Cycle nicht. In diesem Fall muss der Befehl nach jedem Power-Cycle des SMC-Tasters erneut gesendet werden.

Der Befehl **Longpress-Zeit** benötigt mind. Firmware V5.3.9 auf dem SMC-Taster.

Publish — /Pub/Data/Smc

Richtung: Gerät publiziert.

Wenn das Polling für eine Taster-Adresse aktiv ist, wird eine Tasterberührung auf diesem Topic gemeldet:

```
{
  "Content": "/Pub/Data/Smc",
  "Address": "{touched button address}",
  "CommandType": "SET_STATUS",
  "Payload": "POST_PRESSED_STATE"
}
```

LED-Ring und Anzeige setzen

Für den Befehl **LED-Ring und Anzeige setzen** (CommandType `SET_PARAMETER`, Offset `PRE_BUTTON_MODE` oder `POST_BUTTON_MODE`) ist die Payload eine durch / getrennte Zeichenkette:

```
{ENABLED, DISABLED or LONGPRESS_ENABLED}/{Quantity}/{LED ring color}/{LED ring effect}/{Display text}
```

Button-Modus (erstes Feld):

- `ENABLED` — Taster kann bestätigt werden; LED-Ring und Anzeige können gesteuert werden.
- `DISABLED` — Taster kann NICHT bestätigt werden; LED-Ring und Anzeige können gesteuert werden.
- `LONGPRESS_ENABLED` — der Taster geht nach einer bestimmten Berührungsdauer (Standard 2 Sekunden) in den Longpress-Zustand; LED-Ring und Anzeige für diesen Zustand werden über den Offset `POST_BUTTON_MODE` gesetzt.

Quantity — eine Zahl von 1 bis 9999. Sie wird automatisch mittig auf der Anzeige ausgerichtet.

LED-Ring-Farbe:

- `COLOFF`
- `COLRED`
- `COLGREEN`
- `COLBLUE`
- `COLYELLOW`
- `COLMAGENTA`
- `COLCYAN`
- `COLWHITE`

LED-Ring-Effekt:

- `SOLID_RING`
- `FLASH_RING`
- `CONFIRM`
- `POINT_ANIMATED_CLOCK`
- `CIRCLE ANIMATED CLOCK`
- `SOLID_ARROW_UP`, `SOLID_ARROW_DOWN`, `SOLID_ARROW_LEFT`, `SOLID_ARROW_RIGHT`
- `FLASH_ARROW_UP`, `FLASH_ARROW_DOWN`, `FLASH_ARROW_LEFT`, `FLASH_ARROW_RIGHT`
- `ANIMATED_ARROW_UP`, `ANIMATED_ARROW_DOWN`, `ANIMATED_ARROW_LEFT`, `ANIMATED_ARROW_RIGHT`

- Diagonale Pfeile: `SOLID_ARROW_UP_LEFT`, `SOLID_ARROW_UP_RIGHT`, `SOLID_ARROW_DOWN_LEFT`, `SOLID_ARROW_DOWN_RIGHT`, `FLASH_ARROW_UP_LEFT`, `FLASH_ARROW_UP_RIGHT`, `FLASH_ARROW_DOWN_LEFT`, `FLASH_ARROW_DOWN_RIGHT`

HINWEIS

Die diagonalen Pfeileffekte benötigen mind. Firmware V5.3.7 auf dem SMC-Taster.

Display text — max. 4 Zeichen. Einige Zeichen können aufgrund der 7-Segment-Beschränkungen nicht dargestellt werden. Verwenden Sie `@` als Leerzeichen (Beispiele: `@Go@`, `donE`).

Wenn sowohl **Quantity** als auch **Display text** definiert sind, wird **Quantity** verwendet. Wird **Display text** verwendet, erfolgt keine automatische Ausrichtung — der Benutzer muss sich selbst darum kümmern (z. B. eine linksbündige Zahl als `42@@`, eine rechtsbündige Zahl als `@@42`). Wird **Quantity** verwendet, erfolgt die Ausrichtung automatisch.

Beispiele

Senden an den SMC-Taster mit der Adresse `1000`.

LED-Ring-Farbe auf Grün (blinkend) setzen und eine Menge von `42` anzeigen:

```
{
  "Content": "/Set/Data/Smc",
  "Address": "1000",
  "CommandType": "SET_PARAMETER",
  "Offset": "PRE_BUTTON_MODE",
  "Payload": "ENABLED/42/COLGREEN/FLASH_RING"
}
```

LED-Ring-Farbe auf Blau setzen und `donE` im Post-Pressed-/Longpress-Zustand anzeigen:

```
{
  "Content": "/Set/Data/Smc",
  "Address": "1000",
  "CommandType": "SET_PARAMETER",
  "Offset": "POST_BUTTON_MODE",
  "Payload": "ENABLED//COLBLUE/SOLID_RING/donE"
}
```

Taster in den Pre-Pressed-Zustand umschalten:

```
{
  "Content": "/Set/Data/Smc",
  "Address": "1000",
  "CommandType": "SET_STATUS",
  "Offset": "SPECIAL_COMMANDS",
  "Payload": "PRE_PRESSED_STATE"
}
```

Longpress-Zeit auf 1 Sekunde setzen:

```
{
  "Content": "/Set/Data/Smc",
  "Address": "1000",
  "CommandType": "SET_PARAMETER",
  "Offset": "LONGPRESS_TIME",
  "Payload": "100"
}
```

Senden an alle angeschlossenen SMC-Taster — alle Taster in den Pre-Pressed-Zustand umschalten:

```
{
  "Content": "/Set/Data/Smc",
  "Address": "ALL_SENSORS",
  "CommandType": "SET_STATUS",
  "Offset": "SPECIAL_COMMANDS",
  "Payload": "PRE_PRESSED_STATE"
}
```

Health und Status

Publish — /Pub/Health/LedStrip

Richtung: Gerät publiziert.

```
{
  "Content": "/Pub/Health/LedStrip",
  "Firmware": "V0.1.0",
  "LED_STRIP_1": {
    "Length": 42,
    "Active": true,
    "CheckPassed": true
  },
  "LED_STRIP_2": {
    "Length": 42,
    "Active": false,
    "CheckPassed": true
  },
  "LED_STRIP_3": {
    "Length": 42,
    "Active": false,
    "CheckPassed": true
  },
  "LED_STRIP_4": {
    "Length": 0,
    "Active": false,
    "CheckPassed": true
  },
  "LED_STRIP_5": {
    "Length": 0,
    "Active": false,
    "CheckPassed": true
  }
}
```

Der Zustand der an das Board angeschlossenen LED-Streifen. Diese Nachricht wird nach dem Einschalten des Geräts publiziert.

Subscribe — /Get/Health/LedStrip

Richtung: Gerät abonniert. Payload: keine.

Status und Informationen der LED-Streifen anfordern. Das Gerät antwortet auf dem entsprechenden Topic `/Pub/Health/LedStrip`.

Publish — /Pub/Health/SysLog

Richtung: Gerät publiziert.

Payload: Ereignisprotokollierung (Info, Warnung, Fehler usw.), Version, Betriebszeit, IP-Adresse usw.

Das Syslog eines Geräts. Diese Nachricht wird alle 6 h sowie bei einer Anfrage zum Status der LED-Streifen publiziert. Diagnosemeldungen (z. B. erkannte Verdrahtungsfehler, Überlastungen usw.) werden ebenfalls auf diesem Topic publiziert — diese Meldungen sind mit `[DIAGNOSIS]` gekennzeichnet.

Allgemeine Einstellungen

Subscribe — /Set/Config/Common

Richtung: Gerät abonniert.

```
{
  "Content": "/Set/Config/Common",
  "LogLevel": {1 to 6},
  "StartDiagnosis": true
}
```

LogLevel — legt die Log-Stufe fest. Die Standardeinstellung ist 3 (Warning). Folgende Stufen können eingestellt werden:

- 1 – Fatal
- 2 – Error
- 3 – Warning (Standard)
- 4 – Notice (protokolliert z. B. die Antwort, wenn ein LED-Segment aktiviert wird)
- 5 – Info
- 6 – Trace

Log-Meldungen mit der gewählten und niedrigeren Nummer (höherer Schweregrad) werden auf dem Syslog-Topic publiziert. Wird die Log-Stufe z. B. auf 4 gesetzt, publiziert das Gerät Log-Meldungen der Stufen Fatal, Error, Warning und Notice. Die Log-Stufen-Einstellung wird nicht im Flash gespeichert — das Gerät startet immer mit der Standardeinstellung.

StartDiagnosis (nur für die Geräte SEH101 und SEH201) — startet die Diagnose an den Ausgangsklemmen und prüft z. B. auf Verdrahtungsfehler. Mögliche Fehler werden auf dem Syslog-Topic publiziert. Die Diagnose wird beim Start automatisch ausgeführt; dieses Flag löst die Funktion einmalig aus.

Firmware-Update

Subscribe — /Call/FirmwareUpdate

Richtung: Gerät abonniert.

```
{
  "Content": "/Call/FirmwareUpdate",
  "Url": "http://onegrid-
fwupd.germanywestcentral.cloudapp.azure.com:80/ota/firmware_V0.3.9-pythonhB.img"
}
```

Startet den Update-Prozess.

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

HINWEIS

Der SEH201 verwendet Firmware-Dateien mit dem Namen `firmware_V0.x.y-pythonhB_smc.img`.

Zeitverteilung

Subscribe — `/Set/Config/Time`

Richtung: Gerät abonniert. Topics:

```
captron.com/Time
captron.com/{product}/nd/{device-id}/Set/Config/Time
```

```
{
  "Content": "/Set/Config/Time",
  "DateTime": "%Y-%m-%d %H:%M:%S"
}
```

Verteilt die Zeit an jedes Gerät. Zeichenkette im Format `%Y-%m-%d %H:%M:%S`, z. B. `2023-02-28T11:18:30`.

Statussignalisierung

Beim Start wird der Gerätestatus über die angeschlossenen LED-Streifen signalisiert. Um die Installation zu erleichtern (Zuordnung der Streifen zu den Ausgängen), leuchtet an jedem Ausgang eine unterschiedliche Anzahl von LEDs auf: An Ausgang 1 leuchtet eine LED, an Ausgang 2 leuchten zwei LEDs usw. Die Signalisierungsfarben werden bis zum ersten `/Set/Data`-Befehl angezeigt.

Farbe	Status
Red	Ethernet-/WLAN-Verbindungsversuch, keine Ethernet-/WLAN-Verbindung
Yellow	MQTT-Verbindungsversuch, keine MQTT-Verbindung
Blue	Fallback-Mechanismus aktiv (geplant ab V0.2.33/V0.3.6), Konfiguration und/oder Netzwerkverbindung prüfen
Cyan/Green-Blue	Verbindung hergestellt — zum Onboarding-Dienst (falls kein Anwender-MQTT konfiguriert ist) oder zum Anwender-MQTT (falls konfiguriert)

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Diagnosemeldungen (für SEHx01-Geräte)

Diagnosemeldungen werden auf dem Syslog-Topic publiziert.

Fehler	Meldung
Spannung für LEDs außerhalb des Bereichs (4,5 V – 5,5 V), z. B. schwaches USB-Netzteil	[DIAGNOSIS] led voltage outside range
LED-Strom unterhalb des erwarteten Stroms für aktivierte Segmente, z. B. LED-Streifen beschädigt, LEDs außerhalb der Streifenlänge aktiviert	[DIAGNOSIS] expected current not in range
Verdrahtungsfehler oder Überstrom, Details in ERROR_BITS: 1: Verdrahtungsfehler am Pin Data/GND 2, 4: Verdrahtungsfehler am Pin Vled/GND 32: LED-Leistung hat die Grenze des Netzteils erreicht	[DIAGNOSIS] failed: {ERROR_BITS}

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Modbus Interface

Übersicht

Dieses Dokument beschreibt die Modbus-TCP/IP-Schnittstelle zur Steuerung der LED-Streifen und SMC-Taster von SEHx01/SEH11-Geräten. Die Modbus-Schnittstelle arbeitet parallel zur bestehenden MQTT-Steuerung. Beide Schnittstellen können gleichzeitig verwendet werden.

Physikalische Schnittstelle

Parameter	Wert
Protokoll	Modbus
Schnittstelle	TCP/IP
Geräte-ID	1 (konfigurierbar)

Register-Übersicht

Holding Registers (Function 03/06/16)

Der Controller unterstützt 5 LED-Streifen (LED_STRIP_1 bis LED_STRIP_5). Jeder Streifen kann mehrere Segmente enthalten. Pro Streifen sind bis zu 8 Segmente konfigurierbar. Jedes Segment unterstützt bis zu 3 Farben.

Globale Register

Register	Beschreibung	Datentyp	Bereich	R/W
0	Modbus Device ID	UINT16	1-247	R/W
2	Apply/Trigger	UINT16	0/1	W
3	Reserved	-	-	-
4	Reserved	-	-	-

Apply/Trigger (Register 2): Nachdem alle gewünschten Streifen-/Segment-Register beschrieben wurden, muss Register 2 auf 1 gesetzt werden, um die Änderungen zu übernehmen. Das Register wird nach der Anwendung der Änderungen automatisch wieder auf 0 zurückgesetzt.

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Ablehnen

Akzeptieren

Streifen-Register

Jeder Streifen belegt einen Block von 100 Registern. Die Basisadresse für Streifen N (N = 1..5) berechnet sich wie folgt:

$$\text{Basisadresse} = 100 * N$$

Offset	Beschreibung	Datentyp	Bereich	R/W
0	Active	UINT16	0/1	R/W
1	Segment Count	UINT16	0-8	R/W
2-9	Reserved	-	-	-

Segment-Register

Innerhalb eines Streifen-Blocks belegt jedes Segment 11 Register. Die Basisadresse für Segment M (M = 0..7) innerhalb von Streifen N lautet:

$$\text{Segment-Basisadresse} = (100 * N) + 10 + (M * 11)$$

Offset	Beschreibung	Datentyp	Bereich	R/W
0	StartLED	UINT16	0-311	R/W
1	StopLED	UINT16	0-311	R/W
2	Effect	UINT16	siehe Effekte	R/W
3	Speed	UINT16	1-246	R/W
4	Color 1 - Red	UINT16	0-255	R/W
5	Color 1 - Green	UINT16	0-255	R/W
6	Color 1 - Blue	UINT16	0-255	R/W
7	Color 2 - Red	UINT16	0-255	R/W
8	Color 2 - Green	UINT16	0-255	R/W
9	Color 2 - Blue	UINT16	0-255	R/W
10	Color 3 - Red	UINT16	0-255	R/W
11	Color 3 - Green	UINT16	0-255	R/W
12	Color 3 - Blue	UINT16	0-255	R/W

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

SMC-Taster-Steuerung (Holding Registers)

Zusätzlich zu den LED-Streifen können auch die angeschlossenen SMC-Taster über Modbus gesteuert werden. Dies entspricht der bestehenden MQTT-Steuerung (Topic `/Set/Data/Smc`). Beide Wege erzeugen intern denselben SMC-Befehl und können parallel verwendet werden.

Der SMC-Block belegt einen eigenen Registerbereich ab Basisadresse **1000** und besitzt einen **eigenen Apply-Trigger**, unabhängig vom LED-Apply (Register 2).

Register	Beschreibung	Datentyp	Bereich	R/W
1000	SMC Address	UINT16	siehe unten	R/W
1001	SMC CommandType	UINT16	0-2 (siehe unten)	R/W
1002	SMC Offset	UINT16	0-5 (siehe unten)	R/W
1003	Button Mode	UINT16	0-2	R/W
1004	Quantity	UINT16	0-65535	R/W
1005	Color	UINT16	0-7	R/W
1006	Ring Effect	UINT16	siehe Ring-Effekt	R/W
1007	Text char 0	UINT16	0-255 (ASCII)	R/W
1008	Text char 1	UINT16	0-255 (ASCII)	R/W
1009	Text char 2	UINT16	0-255 (ASCII)	R/W
1010	Text char 3	UINT16	0-255 (ASCII)	R/W
1011	Special Command	UINT16	siehe Spezialbefehl	R/W
1012	Button Rotation	UINT16	0-1	R/W
1013	Assign Address	UINT16	siehe unten	R/W
1014-1019	Reserved	-	-	-
1020	SMC Apply/Trigger	UINT16	0/1	W
1100-1228	Hub Polling Channels	UINT16	siehe unten	R/W

Analog zum MQTT-Protokoll werden `CommandType` (Register 1001) und `Offset` (Register 1002) als zwei getrennte Felder geführt. Gemeinsam legen sie fest, welche Payload-Register beim Apply ausgewertet werden. Die Hub Polling Channels belegen einen eigenen Block (Register 1100-1228), der weiter unten beschrieben wird.

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. [Datenschutzerklärung](#).

SMC Address (Register 1000): Zieladresse des Befehls.

Wert	Bedeutung
0	HUB (SEH selbst)
0xFFFF	ALL_SENSORS (Broadcast an alle Taster)
andere	Spezifische Taster-Adresse (Geräteadresse)

SMC CommandType (Register 1001): Spiegelt das Feld `CommandType` des MQTT-Protokolls.

Wert	CommandType
0	- (inaktiv, nichts gesendet)
1	SET_PARAMETER
2	SET_STATUS

Das SMC-Protokoll definiert zusätzlich `GET_STATUS`, `SET_UPDATE`, `GET_INTERNAL_STATUS` und `SET_INTERNAL_STATUS`. Diese sind noch nicht abgebildet und können später ergänzt werden.

SMC Offset (Register 1002): Spiegelt das Feld `Offset` des MQTT-Protokolls.

Wert	Offset
0	(keine / leer)
1	PRE_BUTTON_MODE
2	POST_BUTTON_MODE
3	BUTTON_ROTATE
4	SPECIAL_COMMANDS
5	ADDRESS_TO_BE_ASSIGNED

Gültige Kombinationen: Beim Apply bestimmen `CommandType` und `Offset` gemeinsam, welche Payload-Register ausgewertet werden.

CommandType	Offset	Adresse	Payload-Register
SET_PARAMETER	PRE_BUTTON_MODE	beliebig	Button Mode, Quantity, Color, Ring Effect, Text (1003-1010)
SET_PARAMETER	POST_BUTTON_MODE	beliebig	Button Mode, Quantity, Color, Ring Effect, Text (1003-1010)
SET_PARAMETER	BUTTON_ROTATE	beliebig	Button Rotation (1012)
SET_PARAMETER	(keine)	HUB (0)	Hub Polling Channels (1100-1228)
SET_STATUS	SPECIAL_COMMANDS	beliebig	Special Command (1011)
SET_STATUS	ADDRESS_TO_BE_ASSIGNED	beliebig	Assign Address (1013)

Button Mode (Register 1003):

Wert	Bedeutung
0	DISABLED
1	ENABLED
2	LONGPRESS_ENABLED

Color (Register 1005):

Wert	Bedeutung	Wert	Bedeutung
0	COLOFF	4	COLYELLOW
1	COLRED	5	COLMAGENTA
2	COLGREEN	6	COLCYAN
3	COLBLUE	7	COLWHITE

Ring Effect (Register 1006):

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Wert	Bedeutung	Wert	Bedeutung
0	SOLID_RING	94	ANIMATED_ARROW_UP
1	FLASH_RING	95	ANIMATED_ARROW_DOWN
2	CONFIRM	96	ANIMATED_ARROW_LEFT
84	POINT_ANIMATED_CLOCK	97	ANIMATED_ARROW_RIGHT
85	CIRCLE_ANIMATED_CLOCK	102	SOLID_ARROW_UP_LEFT
86	SOLID_ARROW_UP	103	SOLID_ARROW_UP_RIGHT
87	SOLID_ARROW_DOWN	104	SOLID_ARROW_DOWN_LEFT
88	SOLID_ARROW_LEFT	105	SOLID_ARROW_DOWN_RIGHT
89	SOLID_ARROW_RIGHT	106	FLASH_ARROW_UP_LEFT
90	FLASH_ARROW_UP	107	FLASH_ARROW_UP_RIGHT
91	FLASH_ARROW_DOWN	108	FLASH_ARROW_DOWN_LEFT
92	FLASH_ARROW_LEFT	109	FLASH_ARROW_DOWN_RIGHT
93	FLASH_ARROW_RIGHT		

Special Command (Register 1011): Nur wirksam, wenn CommandType = SET_STATUS und Offset = SPECIAL_COMMANDS.

Wert	Bedeutung
0	PRE_PRESSED_STATE
1	POST_PRESSED_STATE
2	REBOOT
4	SHOW_CURRENT_ADDRESS
5	SELF_TRIGGER
6	LONG_PRESSED_STATE

Text (Register 1007-1010): Bis zu 4 ASCII-Zeichen für die Taster-Anzeige, ein Zeichen pro Register (Register 1007 = erstes Zeichen). Das Zeichen wird im Low-Byte gespeichert; das High-Byte wird ignoriert. Unbenutzte Zeichen (Wert 0) werden intern mit @ aufgefüllt. Nur relevant für Ring-Effekte mit Textanzeige.

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Button Rotation (Register 1012): Nur wirksam, wenn CommandType = SET_PARAMETER und Offset = BUTTON_ROTATE. Legt die Ausrichtung von Anzeige/Taster fest.

Wert	Bedeutung
0	BUTTON_ROTATION_NORMAL
1	BUTTON_ROTATION_UPSIDEDOWN

Assign Address (Register 1013): Nur wirksam, wenn CommandType = SET_STATUS und Offset = ADDRESS_TO_BE_ASSIGNED. Weist dem/den adressierten Taster(n) eine Geräteadresse zu.

Wert	Bedeutung
0	NEXT (automatische Zuweisung der nächsten freien Adresse)
andere	Explizite zuzuweisende Geräteadresse

Hub Polling Channels (Register 1100-1228): Nur wirksam, wenn CommandType = SET_PARAMETER, Offset = (keine) und Address = HUB (0). Legt fest, welche Taster-Adressen der HUB (SEH) aktiv abfragt. Es werden bis zu **128** Adressen unterstützt.

Register	Beschreibung	Datentyp	Bereich
1100	Polling Channel Count	UINT16	0-128
1101	Polling Channel 0	UINT16	Taster-Adresse
1102	Polling Channel 1	UINT16	Taster-Adresse
...	...	...	...
1228	Polling Channel 127	UINT16	Taster-Adresse

Polling-Kanal K befindet sich in Register $1101 + K$ ($K = 0..127$). Register 1100 enthält die Anzahl der gültigen Kanäle. Da eine einzelne Modbus-Anfrage auf ~123 Register begrenzt ist, muss eine lange Kanalliste in mehreren Function-16-Anfragen geschrieben werden, bevor der Apply-Trigger (Register 1020) gesetzt wird.

Die maximale Anzahl tatsächlich nutzbarer Polling-Adressen ist produktabhängig und kann geringer sein als die hier reservierten 128 Register.

SMC Apply/Trigger (Register 1020): Nachdem die SMC-Register beschrieben wurden, muss Register 1020 auf 1 gesetzt werden, um den Befehl an die Taster zu senden. Das Register wird automatisch auf 0 zurückgesetzt. Der SMC-Apply ist unabhängig vom LED-Apply (Register 2).

Input Registers (Function 04) - Status/Nur-Lese

Register	Beschreibung	Datentyp	Bereich
0	Firmware Version Major	UINT16	0-255
1	Firmware Version Minor	UINT16	0-255
2	Firmware Version Patch	UINT16	0-255
3	Number of Configured Strips	UINT16	0-5
4	Strip 1 - Length (LEDs)	UINT16	0-312
5	Strip 2 - Length (LEDs)	UINT16	0-312
6	Strip 3 - Length (LEDs)	UINT16	0-312
7	Strip 4 - Length (LEDs)	UINT16	0-312
8	Strip 5 - Length (LEDs)	UINT16	0-312
9	Power Supply Limit (W×10)	UINT16	0-65535
10	Total Active LEDs	UINT16	0-65535
11	Error Code	UINT16	0-4

Fehlercodes:

Code	Bedeutung
0	Kein Fehler
1	Kein Netzwerk
2	Keine MQTT-Verbindung
3	Keine Konfiguration
4	Normaler Betrieb

SMC-Taster-Status (Input Registers)

Die Zustände der angeschlossenen SMC-Taster können über Input-Register zurückgelesen werden. Dies entspricht der über MQTT auf /Pub/Data/Smc publizierten Taster-Rückmeldung. Der SMC-Statusbereich beginnt bei Input-Register-Adresse **100**.
Wir werden die SMC-Status über unser Webportal (Webster) verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Da Modbus keinen serverseitig initiierten Push kennt, müssen Statusänderungen durch Polling erkannt werden (siehe auch den Abschnitt Hinweise). Um nicht zyklisch den gesamten Statusblock lesen zu müssen, steht der **SMC Event Counter** zur Verfügung.

SMC Event Counter

Register	Beschreibung	Datentyp	Bereich
100	SMC Event Counter	UINT16	0-65535

Der Zähler wird bei jeder SMC-Taster-Statusänderung erhöht und läuft bei 65535 über. Ein Client kann dieses einzelne Register kostengünstig abfragen und die Taster-Statusblöcke nur dann erneut lesen, wenn sich der Wert geändert hat. Die Register 101-109 sind reserviert.

Taster-Status-Register

Die Anzahl der abgebildeten Taster entspricht `MAX_SMC_COUNT`. Dieser Wert hängt vom Build-Target ab (z. B. 5 bei Standardvarianten, bis zu 128 bei bestimmten Geräten). Jeder Taster belegt einen Block von 4 Registern. Die Basisadresse für Taster K ($K = 0 \dots \text{MAX_SMC_COUNT}-1$) berechnet sich wie folgt:

$$\text{Taster-Basisadresse} = 110 + (K * 4)$$

Offset	Beschreibung	Datentyp	Bereich
0	Button Address	UINT16	0-65535
1	Press Count	UINT16	0-65535
2	Press State	UINT16	siehe unten
3	Reserved	-	-

Eine Button Address von 0 kennzeichnet einen unbenutzten/leeren Slot.

Press State (Offset 2):

Wert	Bedeutung
0	PRE_PRESSED_STATE
1	POST_PRESSED_STATE
3	LONGPRESS_STATE

Taster	Adresse	Press Count	Press State
0	110	111	112
1	114	115	116
2	118	119	120
...	...	...	...
K	$110 + K \times 4$	$111 + K \times 4$	$112 + K \times 4$

Verfügbare LED-Effekte

Effekt-ID	Name	Beschreibung
1	Static	Alle LEDs leuchten in konstanter Farbe
2	Blink	Ein-/Aus-Blinken
3	Flash	Kurze, helle Blitze
4	Leading Lines	Vorwärts laufende Linien
5	Leading Lines Reverse	Rückwärts laufende Linien
6	Multipicker	Durchlaufen mehrerer Farben
7	Pick	Animation von der Mitte nach außen
8	Place	Animation von außen zur Mitte
200	Knight Rider	Hin- und herlaufende Balkenanimation

Register-Zuordnung - Beispiel

Streifen 1, Segment 0 aktivieren (green, static)

Folgende Register müssen beschrieben werden (Function 16 - Write Multiple Registers):

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Register	Wert	Beschreibung
100	1	Strip 1 Active = true
101	1	Strip 1 Segment Count = 1
110	0	Segment 0: StartLED = 0
111	20	Segment 0: StopLED = 20
112	1	Segment 0: Effect = Static
113	200	Segment 0: Speed = 200
114	0	Segment 0: Color1 R = 0
115	150	Segment 0: Color1 G = 150
116	0	Segment 0: Color1 B = 0
117	0	Segment 0: Color2 R = 0
118	0	Segment 0: Color2 G = 0
119	0	Segment 0: Color2 B = 0
120	0	Segment 0: Color3 R = 0
121	0	Segment 0: Color3 G = 0
122	0	Segment 0: Color3 B = 0
2	1	Apply/Trigger

Dies entspricht der folgenden MQTT-Nachricht:

```
{
  "LED_STRIP_1": {
    "Active": true,
    "Segments": [
      {
        "StartLED": 0,
        "StopLED": 20,
        "Effect": 1,
        "Speed": 200,
        "Colors": [{ "R": 0, "G": 150, "B": 0 }]
      }
    ]
  }
}
```

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

SMC-Taster: PRE_BUTTON_MODE setzen (alle Taster, green, blinkender Ring)

Folgende Register müssen beschrieben werden (Function 16 - Write Multiple Registers):

Register	Wert	Beschreibung
1000	0xFFFF	Address = ALL_SENSORS (Broadcast)
1001	1	CommandType = SET_PARAMETER
1002	1	Offset = PRE_BUTTON_MODE
1003	1	Button Mode = ENABLED
1004	5	Quantity = 5
1005	2	Color = COLGREEN
1006	1	Ring Effect = FLASH_RING
1020	1	SMC Apply/Trigger

Dies entspricht der folgenden MQTT-Nachricht:

```
{
  "Address": "ALL_SENSORS",
  "CommandType": "SET_PARAMETER",
  "Offset": "PRE_BUTTON_MODE",
  "Payload": "ENABLED/5/COLGREEN/FLASH_RING/@@@@"
}
```

SMC-Taster: Spezialbefehl auslösen (PRE_PRESSED_STATE, alle Taster)

Register	Wert	Beschreibung
1000	0xFFFF	Address = ALL_SENSORS (Broadcast)
1001	2	CommandType = SET_STATUS
1002	4	Offset = SPECIAL_COMMANDS
1011	0	Special Command = PRE_PRESSED_STATE
1020	1	SMC Apply/Trigger

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzutzerklärung](#).

Dies entspricht der folgenden MQTT-Nachricht:

```
{
  "Address": "ALL_SENSORS",
  "CommandType": "SET_STATUS",
  "Offset": "SPECIAL_COMMANDS",
  "Payload": "PRE_PRESSED_STATE"
}
```

SMC-Taster: Taster-Rotation setzen (alle Taster, kopfüber)

Register	Wert	Beschreibung
1000	0xFFFF	Address = ALL_SENSORS (Broadcast)
1001	1	CommandType = SET_PARAMETER
1002	3	Offset = BUTTON_ROTATE
1012	1	Button Rotation = BUTTON_ROTATION_UPSIDEDOWN
1020	1	SMC Apply/Trigger

Dies entspricht der folgenden MQTT-Nachricht:

```
{
  "Address": "ALL_SENSORS",
  "CommandType": "SET_PARAMETER",
  "Offset": "BUTTON_ROTATE",
  "Payload": "BUTTON_ROTATION_UPSIDEDOWN"
}
```

SMC-Taster: Adresse zuweisen (alle Taster, Adresse 0x1000)

Register	Wert	Beschreibung
1000	0xFFFF	Address = ALL_SENSORS (Broadcast)
1001	2	CommandType = SET_STATUS
1002	5	Offset = ADDRESS_TO_BE_ASSIGNED
1013	0x1000	Assign Address = 0x1000
1020	1	SMC Apply/Trigger

Wir verwenden Cookies für Analysen, um unsere Website zu verbessern. Weitere Informationen finden Sie in unserer [Datenschutzerklärung](#).

Dies entspricht der folgenden MQTT-Nachricht:

```
{
  "Address": "ALL_SENSORS",
  "CommandType": "SET_STATUS",
  "Offset": "ADDRESS_TO_BE_ASSIGNED",
  "Payload": "1000"
}
```

Setzen Sie Register 1013 auf 0, um automatisch die nächste freie Adresse zuzuweisen (MQTT-Payload "NEXT").

SMC-HUB: Polling-Kanäle konfigurieren (Taster 0x1000-0x1003 abfragen)

Register	Wert	Beschreibung
1000	0	Address = HUB
1001	1	CommandType = SET_PARAMETER
1002	0	Offset = (keine)
1100	4	Polling Channel Count = 4
1101	0x1000	Polling channel 0
1102	0x1001	Polling channel 1
1103	0x1002	Polling channel 2
1104	0x1003	Polling channel 3
1020	1	SMC Apply/Trigger

Dies entspricht der folgenden MQTT-Nachricht:

```
{
  "Address": "HUB",
  "CommandType": "SET_PARAMETER",
  "Offset": "",
  "Payload": "1000/1001/1002/1003"
}
```

Adressberechnung - Kurzreferenz

Strip N Base Address: $100 * N$ ($N = 1..5$)
 Strip N Active: $100 * N + 0$
 Strip N Segment Count: $100 * N + 1$
 Segment M Base Address: $100 * N + 10 + (M * 11)$ ($M = 0..7$)

SMC block (holding): selector 1000-1002, payload 1003-1013, Apply = 1020
 SMC hub polling (holding): count 1100, channel K = 1101 + K ($K = 0..127$)
 SMC event counter (input): 100
 SMC status (input): button K base = 110 + (K * 4) ($K = 0 ..$
 MAX_SMC_COUNT-1)

Streifen	Active	Seg.-Anzahl	Seg 0	Seg 1	Seg 2	Seg 3
1	100	101	110	121	132	143
2	200	201	210	221	232	243
3	300	301	310	321	332	343
4	400	401	410	421	432	443
5	500	501	510	521	532	543

Hinweise

- **Paralleler MQTT-/Modbus-Betrieb:** Beide Schnittstellen schreiben auf dieselben LED-Streifen. Der zuletzt empfangene Befehl (ob MQTT oder Modbus) bestimmt den aktuellen Zustand.
- **Leistungsgrenze:** Die Leistungsgrenze gilt global. Segmente, die die Leistungsgrenze überschreiten würden, werden nicht aktiviert (identisches Verhalten wie bei MQTT).
- **StopLED >= StartLED:** Ist StopLED < StartLED, wird StopLED automatisch auf StartLED gesetzt.
- **Farben:** Die Summe aus R+G+B pro Farbe ist auf maximal 180 begrenzt (Strombegrenzung pro LED).
- **Ungenutzte Farben:** Setzen Sie ungenutzte Farbreister (Color 2, Color 3) auf 0/0/0.
- **Apply-Register:** Änderungen an Streifen-/Segment-Registern werden erst wirksam, wenn 1 in Register 2 (Apply) geschrieben wird. Dies ermöglicht eine atomare Konfiguration mehrerer Streifen/Segmente.
- **SMC-Steuerung:** Der SMC-Block (ab Register 1000) besitzt einen eigenen Apply-Trigger (Register 1020), unabhängig vom LED-Apply (Register 2). LED- und SMC-Befehle können somit getrennt ausgelöst werden.
- **SMC parallel zu MQTT:** Modbus- und MQTT-SMC-Befehle erzeugen intern denselben SMC-Befehl und werden in dieselbe Sendewarteschlange eingereiht. Der zuletzt empfangene Befehl bestimmt den Taster-Zustand.
- **SMC-Adressierung:** 0xFFFF adressiert alle Taster (Broadcast), 0 den HUB (SEH selbst), jeder andere Wert einen bestimmten Taster.

- **SMC-Status:** Die Input-Register für den Taster-Status (ab Adresse 110) spiegeln die zuletzt empfangene Taster-Rückmeldung wider. Eine Button Address von 0 kennzeichnet einen unbenutzten Slot.
- **Benachrichtigung bei Statusänderung:** Modbus verfügt über keinen serverseitig initiierten Push; der Server beantwortet nur Anfragen. Statusänderungen werden durch Polling des SMC Event Counter (Register 100) erkannt; die Taster-Statusblöcke werden bei einer Änderung erneut gelesen. Für echte Push-Benachrichtigungen nutzen Sie den parallel laufenden MQTT-Kanal (`/Pub/Data/Smc`).