go-eCharger Modbus API einrichten und Register lesen

Die go-eCharger Modbus API gibt Messwerte und Steuerbefehle über Modbus TCP frei und braucht dafür eine vorherige Aktivierung in App oder Schnittstelle. Nach der Modbus-Dokumentation von go-e auf GitHub, abgerufen am 15.11.2026, nutzt sie nach Freigabe den TCP Port 502 und stellt Halte Register zum Lesen und Schreiben sowie Eingabe Register nur zum Lesen bereit. Die Konfiguration mit Unit ID 1, nullbasierter Adressierung und Big Endian beschreibt Modbus Connect, abgerufen am 15.11.2026, als Standard. Damit eignet sich die Schnittstelle für Haussteuerung, Lastmanagement und Überschussladen, sofern Register, Funktionen und Schreibgrenzen beachtet werden.

go-eCharger Modbus API aktivieren: was gibt den Zugriff frei?

Die go-eCharger Modbus API bleibt bis zur Freigabe in App oder Schnittstelle gesperrt und antwortet erst danach auf dem TCP Port 502. Die Freigabe liegt in der App unter Internet und erweiterte Einstellungen bei Modbus und lässt sich alternativ über Schlüssel der http API setzen. Eine saubere Prüfung folgt dem gleichen Testverfahren: erst Verbindung und Freigabe kontrollieren, dann Register lesen, erst danach schreibend eingreifen. Die ältere Dokumentation mit Version 1.0 vom 01.10.2020 zeigt die lange Linie dieser Schnittstelle, die aktuelle Modbus-Dokumentation beschreibt Aktivierung, Funktionen und Register im Detail.

Die Integrationsanleitung in der neoom Wissensdatenbank, aktualisiert am 13.11.2026, beschreibt den Weg über Einstellungen und API Einstellungen mit Zulassen der lokalen Modbus TCP API und Setzen der Vertauschungsoption. Dort werden Lesen von Status, Firmware, Wirkleistung, Wirkleistungsfaktor, Gesamtenergie, Ladevorgangsenergie, Fehler, Spannungen und Strömen sowie Lesen und Schreiben von Start und Stopp, maximalem Ladestrom, Ladefreigabe und Phasennutzung unterstützt. Die Supportseite von go-e bündelt Anleitungen, API Dokumentation und Downloads als Dateien und Videos und dient als Ablage für Produktunterlagen. Für den Betrieb zählt damit nicht die Datei selbst, sondern die passende Freigabe plus passende Registerwahl.

Welche Funktionen stellt die Schnittstelle für Lesen und Schreiben bereit?

Die Schnittstelle unterstützt Funktion 3 zum Lesen der Halte Register, Funktion 4 zum Lesen der Eingabe Register und Funktion 16 zum Schreiben mehrerer Register. Das Schreiben eines einzelnen Registers gelingt nur mit Funktion 16, Spulenfunktionen werden nicht unterstützt und bleiben für Steuerungsaufgaben außen vor. Die folgende Einordnung fasst die in der Modbus-Dokumentation beschriebenen Funktionen mit Stand vom 15.11.2026 zusammen.

Aufgabe | Funktion | Zugriff

Halte Register lesen | 3 | lesend
Eingabe Register lesen | 4 | lesend
Mehrere Halte Register schreiben | 16 | schreibend

Einzelnes Register direkt schreiben | 16 statt Einzelfunktion | schreibend über Mehrfachschreibung
Spulen lesen und schalten | nicht unterstützt | kein Zugriff

Die Trennung ist für die Praxis wichtig, weil Messwerte nur gelesen und Vorgaben nur über Halte Register geschrieben werden. Wer Lesezugriff und Schreibzugriff vermischt, erhält Fehler oder keine Wirkung an der Station. Die Konfiguration mit Port 502 und Unit ID 1 bleibt dabei fest, Adressierung und Wortreihenfolge folgen dem Standard mit nullbasierter Adressierung und Big Endian. Damit passt die Schnittstelle zu gängigen Steuerungen, sofern die Steuerung Funktion 3, Funktion 4 und Funktion 16 sauber umsetzt.

Welche Halte und Eingabe Register steuern Laden und Messung?

Die wichtigsten Register verknüpfen Ladefreigabe, Stromvorgabe, Fahrzeugstatus, Kabelerkennung, Fehler und Gesamtenergie in einer lesbaren Struktur. Halte Register nehmen Vorgaben auf, Eingabe Register liefern Messung und Status ohne Schreibweg. Die folgende Auswahl zeigt die für Betrieb und Fehlersuche tragenden Adressen mit Stand vom 15.11.2026.

Register | Name | Inhalt

40201 | ALLOW | Ladefreigabe
40300 | AMPERE_VOLATILE | Stromvorgabe ohne dauerhafte Speicherung im Bereich von 6 bis 32
40301 | AMPERE_EEPROM | Stromvorgabe mit Speicherung im Bereich von 6 bis 32
40207 | LED_BRIGHTNESS | Helligkeit im Bereich von 0 bis 255, wobei 0 aus bedeutet
30101 | CAR_STATE | Fahrzeugstatus mit Werten von 0 bis 4
30102 | PP_CABLE | Kabelcodierung mit 0 für kein Kabel und 13 bis 32 für Codierung
30108 | ERROR | Fehler mit 1 für Schutzschalter, 3 für Phasenstörung, 8 für Erdungserkennung und 10 für sonstige Fehler
128 | ENERGY_TOTAL | gesamte geladene Energie in 0.1 Kilowattstunden

Die Stromwerte werden als Rohwert in 0.1 Ampere gemeldet und brauchen daher eine saubere Skalierung in der Steuerung. Passend zum go-eCharger HOME+ lassen sich damit Ladestrom, Freigabe und Anzeige getrennt behandeln, ohne an der Station selbst zu schalten. Die Statuswerte von 0 bis 4 unterscheiden Bereitschaft ohne Fahrzeug, laufende Ladung, Warten auf Fahrzeug und beendete Ladung bei noch verbundenem Fahrzeug. Die Fehlerwerte trennen Schutzschalter, Phasenstörung, Erdungserkennung und sonstige interne Fehler und geben der Fehlersuche eine klare Richtung.

Wie läuft die Anbindung an Steuerung und Überschussladen?

Die Anbindung gelingt über festen Port, feste Unit ID und klare Trennung von lesbaren Messwerten und schreibbaren Vorgaben. Die Steuerung liest zyklisch Status und Energie, setzt Freigabe und Stromvorgabe nur bei Bedarf und prüft die Wirkung über den nächsten Lesezyklus. Die Übersicht der go-eCharger hilft bei der Einordnung der Station im Haus, die eigentliche Logik liegt in der Steuerung und nicht in der Datei. Schreibende Zugriffe gehören nur auf Halte Register, lesende Zugriffe auf beide Bereiche, Spulenbefehle entfallen vollständig.

Für Überschussladen und Lastmanagement zählen vor allem Gesamtenergie, Ladevorgangsenergie, Wirkleistung, Wirkleistungsfaktor, Spannungen und Ströme auf der Leseseite sowie Start und Stopp, maximaler Ladestrom, Ladefreigabe und Phasennutzung auf der Schreibseite. Die RFID Karte zur Freigabe bleibt auslesbar und ordnet Ladevorgänge einer Karte zu. Die Helligkeit lässt sich unabhängig vom Laden anpassen und beeinflusst keine Messung. Damit bleibt die Station ruhig im Netz, solange Abfragezyklus, Schreibgrenzen und Freigabelogik zusammenpassen.

go-eCharger Modbus API Häufige Fragen

Was schaltet die go-eCharger Modbus API für Lesen und Schreiben frei?

Die go-eCharger Modbus API wird erst durch Freigabe in App oder Schnittstelle aktiv und nutzt danach den TCP Port 502. Danach erlaubt Funktion 3 das Lesen der Halte Register, Funktion 4 das Lesen der Eingabe Register und Funktion 16 das Schreiben mehrerer Register. Die Konfiguration nutzt Unit ID 1 mit nullbasierter Adressierung und Big Endian. Ohne Freigabe antwortet die Schnittstelle nicht sinnvoll, daher beginnt jede Anbindung mit Prüfung von Verbindung und Freigabe.

Welche Register bedient die go-eCharger Modbus API für Strom und Freigabe?

Die go-eCharger Modbus API nutzt Register 40201 für die Ladefreigabe und Register 40300 sowie Register 40301 für die Stromvorgabe. Der Bereich von 6 bis 32 gilt für beide Stromregister, wobei das volatile Register ohne dauerhafte Speicherung arbeitet und das andere Register mit Speicherung arbeitet. Register 40207 stellt die Helligkeit im Bereich von 0 bis 255 ein. Diese Trennung hält Betrieb, Anzeige und Vorgabe sauber auseinander.

Welche Messwerte liefert die go-eCharger Modbus API aus dem Ladevorgang?

Die go-eCharger Modbus API liefert über Register 30101 den Fahrzeugstatus mit Werten von 0 bis 4 und über Register 30102 die Kabelcodierung. Dabei bedeutet 0 kein Kabel und 13 bis 32 die Codierung des angeschlossenen Kabels. Register 30108 meldet 1 für Schutzschalter, 3 für Phasenstörung, 8 für Erdungserkennung und 10 für sonstige Fehler. Das Registerpaar 128 liefert die Gesamtenergie in 0.1 Kilowattstunden, Stromwerte kommen in 0.1 Ampere.

Wofür eignet sich die go-eCharger Modbus API in Haussteuerung und Energiemanagement?

Die go-eCharger Modbus API eignet sich für Haussteuerung, Lastmanagement und Überschussladen mit klarer Trennung von Lesen und Schreiben. Nach der Integrationsanleitung in der neoom Wissensdatenbank, aktualisiert am 13.11.2026, lassen sich Status, Leistung, Energie, Fehler, Spannungen und Ströme lesen sowie Start und Stopp, maximaler Ladestrom, Freigabe und Phasennutzung lesen und schreiben. Die Steuerung setzt Vorgaben sparsam und prüft die Wirkung über den nächsten Lesezyklus.