Anleitung COMEXIO Modbus-Bridge

Erläuterung Konfiguration

Baudrate: Die Übertragungsgeschwindigkeit der Busdaten. In der Regel in der Anleitung des Endgeräts zu entnehmen. Es kann für jeweils eine RS-Bridge nur eine Baudrate geben.

Datenbits: Hier wird definiert, wie viele relevante Bits ein Zeichen hat. Es ist ebenfalls der Anleitung des Endgeräts zu entnehmen.

Stopbits: Markiert das Ende des Zeichens. Können Sie ebenfalls in der Anleitung des Endgeräts entnehmen.

Parität: Bei der Nachrichtenübertragung eine einfache Methode der Fehlererkennung. In der Anleitung des Endgeräts zu entnehmen.

Stopbyte: Das Zeichen, dass das Ende einer Nachricht markiert. Angegeben als Hexadezimalzeichen des ASCII-Zeichen (siehe ASCII Tabelle).

Minimale Zeit zwischen Nachrichten: Falls kein „Stopbyte“ gegeben ist, muss eine minimale Zeit zwischen Nachrichten vergehen, um das Ende einer Nachricht zu erkennen. Hier sind z. B. 100ms ausreichend.

Rechenbeispiel:

9600 Baud = 9600 Zeichen pro Sekunde

1 Zeichen wird in ca. 1/9600 s übertragen

Es muss das Zweieinhalbfache an Zeit für ein Zeichen vergehen, bevor die nächste Nachricht gesendet werden kann

*1/9600 = 0,1041 ms * 2,5 → 0,25 ms wäre theoretisch ausreichend*

Neues Gerät anlegen

Name: Der Name des Geräts z. B. Klimaanlage.

Beschreibung: Beschreibung des Geräts in der Geräteansicht.

Feste Geräteadresse: Bei Modbus RTU wird vorausgesetzt, dass die Geräteadresse am Anfang jeder Nachricht angehängt ist. Wenn Sie die Adresse auswählen, wird bei jeder Nachricht automatisch diese in der Nachricht an erster Stelle eingefügt.

Ist der Haken nicht gesetzt, muss die Adresse, sofern vorhanden, manuell bei jeder Nachricht gesetzt werden.

CRC-TYP: Eine weitere Fehlererkennungsmetode, die vom Modbus-Protokoll vorausgesetzt wird. Es werden zwei Byte am Ende jeder Nachricht angehängt. Die Endgeräte gleichen die CRC Bytes ab um zu erkennen, ob ein Fehler bei der Übertragung zu Stande kam. Deaktivieren, wenn Endgeräte keine CRC Kontrolle verfügen. Fehlende CRC kann zum ignorieren aller ankommenden Nachrichten beim Endgerät führen.

Befehle anlegen:

Name: Name des Befehls z. B. An/Aus

Beschreibung: Beschreibung des Befehls

Digital: Auswahlmöglichkeit ob der erzeugte Datenpunkt Digital oder Analog ist

COMEXIO Eingang: Hier wird festgelegt ob es sich um einen Ausgang oder einen Eingang handelt.

- Ausgang: COMEXIO sendet einen Befehl an das Endgerät (z. B. Einschaltbefehl oder Anforderung eines Status)
- Eingang: COMEXIO erhält Antwort eines Endgeräts (z. B. Statusrückmeldung)

Antwortlänge (optional): Zusätzliche Sicherheit um ankommende Nachrichten im Vorfeld zuordnen zu können (Standard=0 eingestellt lassen, um die Funktion zu ignorieren).

Antwortmuster (optional): Sollte die Antwortlänge mit der gesendeten Nachricht übereinstimmen, so kann im Antwortmuster überprüft werden, ob diese Nachricht zum Befehl passt.

Das Antwortmuster ist ein LUA Pattern.

Siehe Beispiel 2: Befehl 2: Interpretieren

Zuordnung: Eine eindeutige Zuordnung von Anforderung zu Interpreter und umgekehrt. Zwingend erforderlich.

Funktionscode: Eine Auswahl an vorgefertigten Modbus Funktionscodes

- Kein: Der Befehl muss selbst geschrieben werden.
- Lesen Holding: Standard Lesebefehl für Holding Register.
- Lesen Input: Lesebefehl für Input Register.

- Schreiben Coil: Es wird in ein Coil geschrieben. Kann nur An (0xFF00) oder Aus (0x0000) sein. Wird verwendet, wenn man der Anleitung entnehmen kann, dass in den Coil geschrieben werden soll.
- Schreiben Register: Standard Schreibbefehl für Holding Register.
- Schreiben mehrere: Schreibbefehl für mehrere Holding Register. Das Endgerät muss den Schreibbefehl für mehrere Register unterstützen.

Registeradresse: Das jeweilige Register, welches ausgelesen oder beschrieben wird. Ist der Anleitung des Endgeräts zu entnehmen.

Anzahl Register: Relevante Angabe für Funktionscode „Lesen Holding“, „Lesen Input“ und „Schreiben mehrere“. Gibt an, wie viele Register ab der „Registeradresse“ gelesen oder beschrieben werden.

Daten: Nur relevant für Schreibbefehle. Man kann entweder einen fixen Wert in das Register schreiben (Fixwert) oder über die COMEXIO Logik einen Wert übergeben, der dann in das Register geschrieben wird (Eingang).

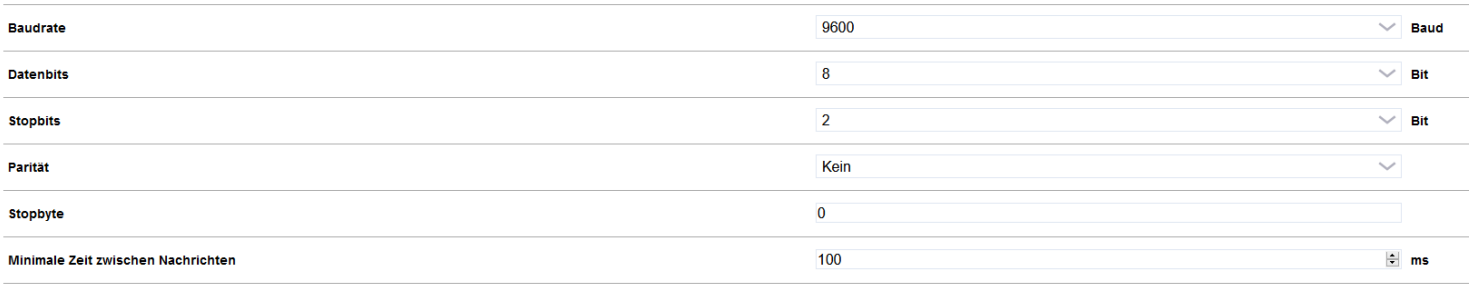
Der Fixwert kann in HEX in der darunter liegenden Zeile eingetragen werden.

Interpretercode: Der resultierende Code. Muss mit „function rs_interpreter (v)“ beginnen und mit „end“ enden.

Achtung: Manuelle Codeänderungen werden bei verändern der Auswahlfelder in der Befehlsmaske kommentarlos verworfen.

Beispiel:

Es handelt sich um eine Modbus RTU (Slave)-Schnittstelle und diese wird an eine RS485-Busleitung angeschlossen. Geführt wird eine 8N2- (8N1-kompatible)-Kommunikation (8 Datenbits, keine Parität und 2 Stopp-Bits) mit mehreren verfügbaren Baudraten: 2400 Bd, 9600 Bd (Vorgabe), 19200 Bd und 57600 Bd).



Screenshot aus der Modbus-Bridge (Konfiguration)

Screenshot aus der Modbus-Bridge (Gerät anlegen)

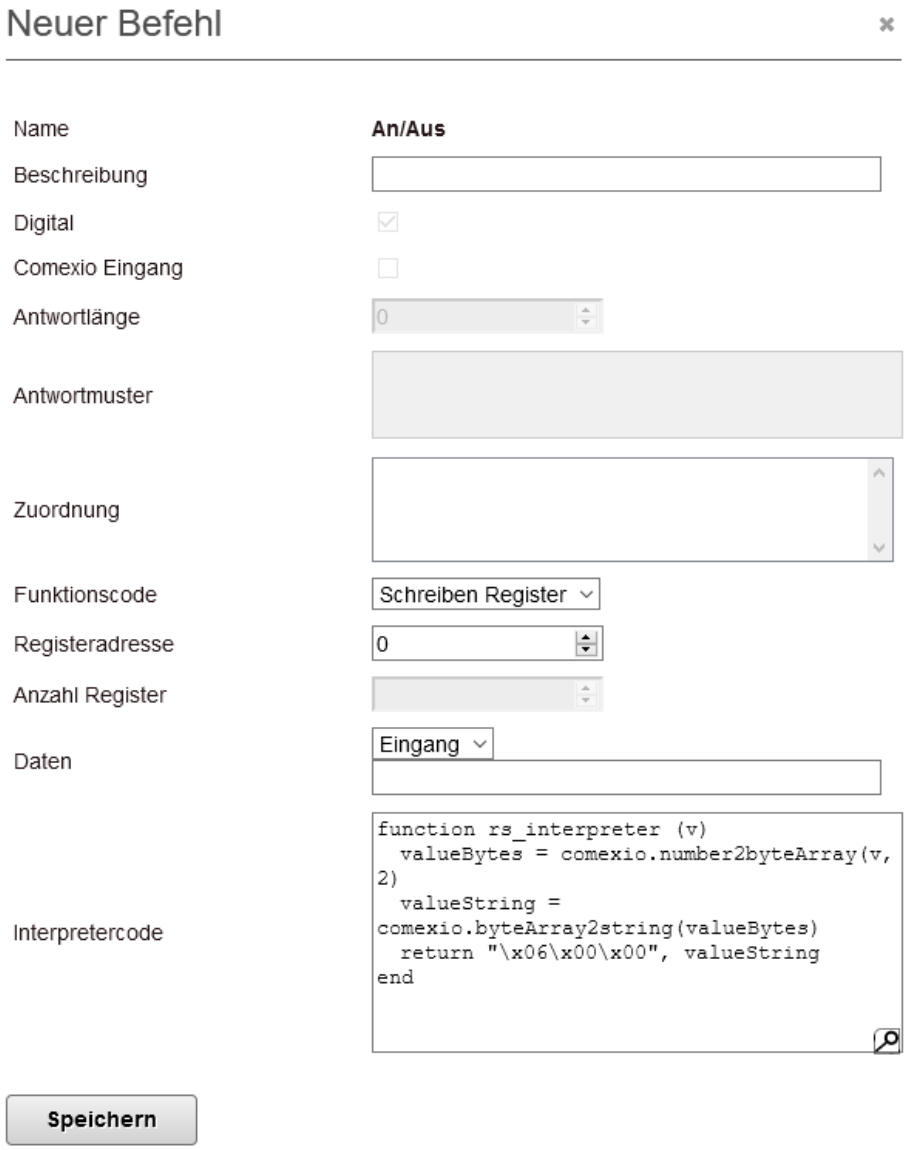
Steuer- und Statusregister (Endgerät):

Registeradresse (Protokolladresse)	Lesen/Schreiben	Beschreibung
0	L/S	Raumklimagerät Ein/Aus • 0: Aus • 1: Ein
1	L/S	Raumklimagerät Betriebsart *1 • 0: Auto • 1: Heizen • 2: Trocknen/Luftentfeuchten • 3: Gebläse • 4: Kühlen
2	L/S	Raumklimagerät Gebläsestufe *1 • 0: Auto • 1: Niedrig • 2: Medium 1 • 3: Medium 2 • 4: Hoch
3	L/S	Raumklimagerät Ausblasrichtung (Vane) Position *1 • 0: Auto • 1: Horizontal • 2: Position 2 • 3: Position 3 • 4: Position 4 • 5: Vertikal • 6: Swing
4	L/S	Raumklimagerät Soll-Raumtemperatur *2*3 • 16–32 °C (°C/×10 °C) • 60–90 °F
5	L	Raumklimagerät Ist-Raumtemperatur *2*3 • 10–38 °C (°C/×10 °C) • 50–100 °F

Achtung: Die Adresse des Endgeräts wurde beim Anlegen fest definiert und ändert sich auch nicht mehr. Die Adresse wird von COMEXIO automatisch bei jeder Nachricht an erster Stelle angehängt. Aus diesem Grund taucht die Adresse im Code nicht auf.

Beispiel 1: Klimagerät einschalten

Das Gerät wurde erfolgreich konfiguriert und angelegt. Der Befehl um die Klimaanlage ein- bzw. auszuschalten sieht so aus.



function rs_interpreter (v)

Die Kopfzeile der Funktion.

valueBytes = comexio.number2byteArray(v,2)

Die vom COMEXIO System übergebene Zahl wird in ein Array mit einzelnen Bytes aufgespalten. Durch die Klammer **(v,2)** wird die Zahl in mindestens zwei Bytes aufgeteilt

valueString = comexio.byteArray2string(valueBytes)

Da ein Array nicht einfach ausgegeben werden kann, muss das Array in einen String umgewandelt werden.

return "\x06\x00\x00", valueString

Der String wird ausgegeben. Bestehend aus drei + die Länge von „valueString“ Bytes (besteht hier aus mindestens zwei Bytes).

1. Der Funktionscode x06 → Schreiben Register
2. Die Adresse des Registers bestehend aus zwei Byte (Byte 2 und 3)
3. Die Adresse des Registers bestehend aus zwei Byte (Byte 2 und 3)
4. Der Wert der geschrieben werden soll. Hier mindestens zwei Byte bestehend aus „valueString“

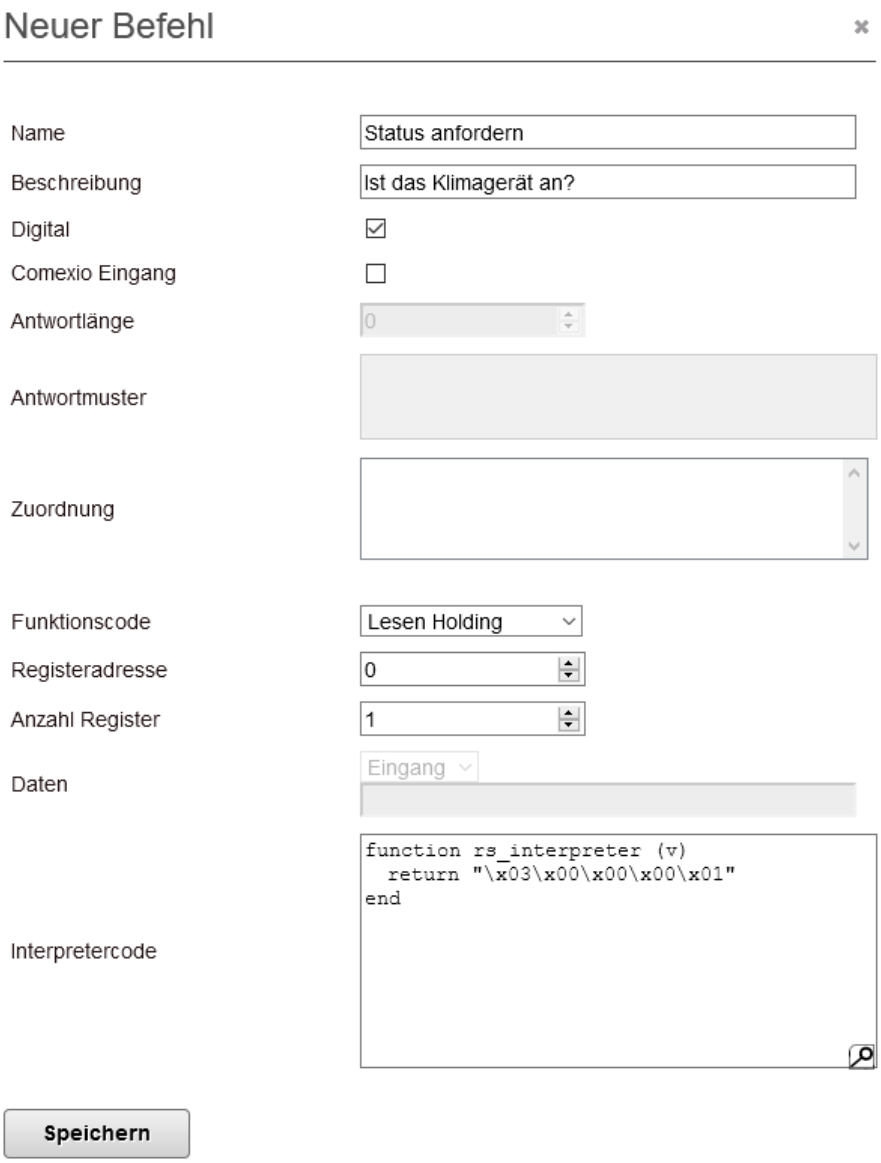
end

Das Ende der Funktion.

Beispiel 2: Zustand auslesen

Um den Zustand der Klimaanlage auslesen zu können, müssen zwei Befehle angelegt werden. Der 1. Befehl ist die Anforderung des Status und der 2. Befehl dient zum Interpretieren der Antwort.

Befehl 1: Anfordern

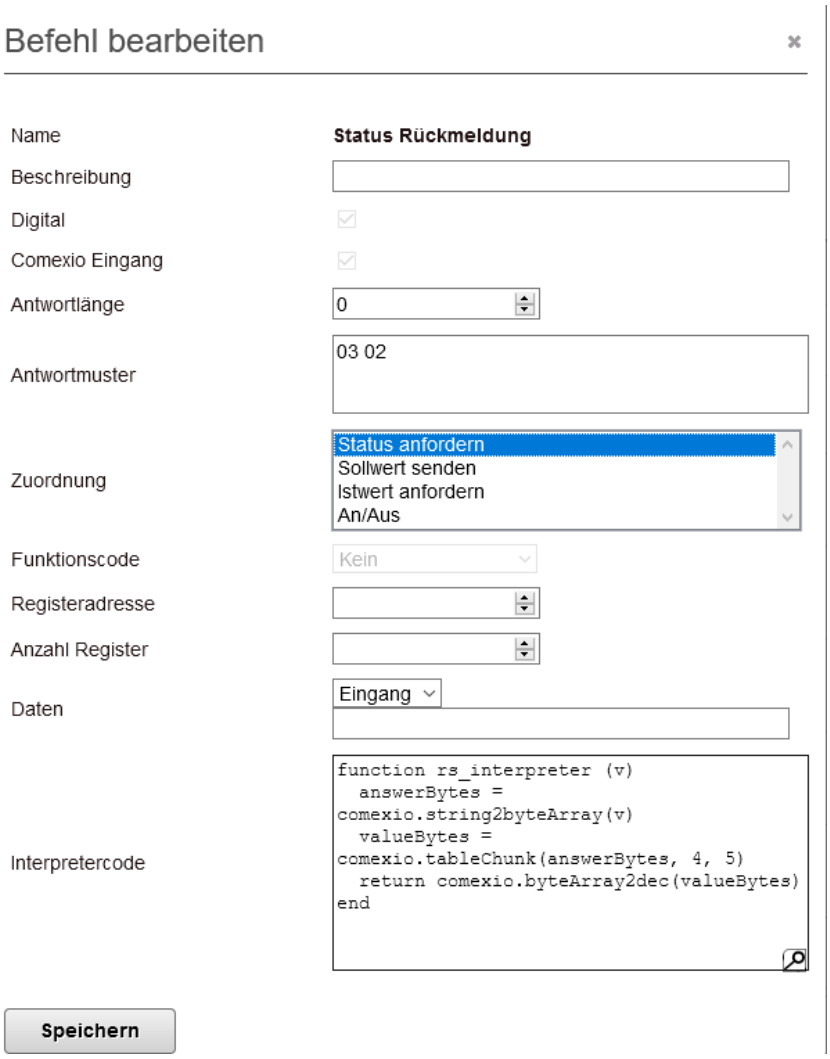


return "\x03\x00\x00\x00\x01"

Der Ausgabewert bestehend aus fünf Byte.

1. Der Funktionscode in diesem Fall x03 → Holding Register lesen
2. Die Adresse des Registers bestehend aus zwei Byte (Byte 2 und 3)
3. Die Adresse des Registers bestehend aus zwei Byte (Byte 2 und 3)
4. Die Anzahl der Register die gelesen werden sollen bestehend aus zwei Byte (Byte 4 und 5)
5. Die Anzahl der Register die gelesen werden sollen bestehend aus zwei Byte (Byte 4 und 5)

Befehl 2: Interpretieren



Antwortmuster: In der Nachricht vom Endgerät muss die Bytefolge „03 02“ vorkommen. Die „03“ gibt den Funktionscode der Anforderung an und „02“ die Länge der gesendeten Daten.

Zuordnung: Dieser Befehl ist dem Befehl „Status anfordern“ eindeutig zugeordnet.

answerBytes = comexio.string2byteArray(v)

Die Antwort des Geräts wird in ein Array mit den einzelnen Byte Werten geschrieben.

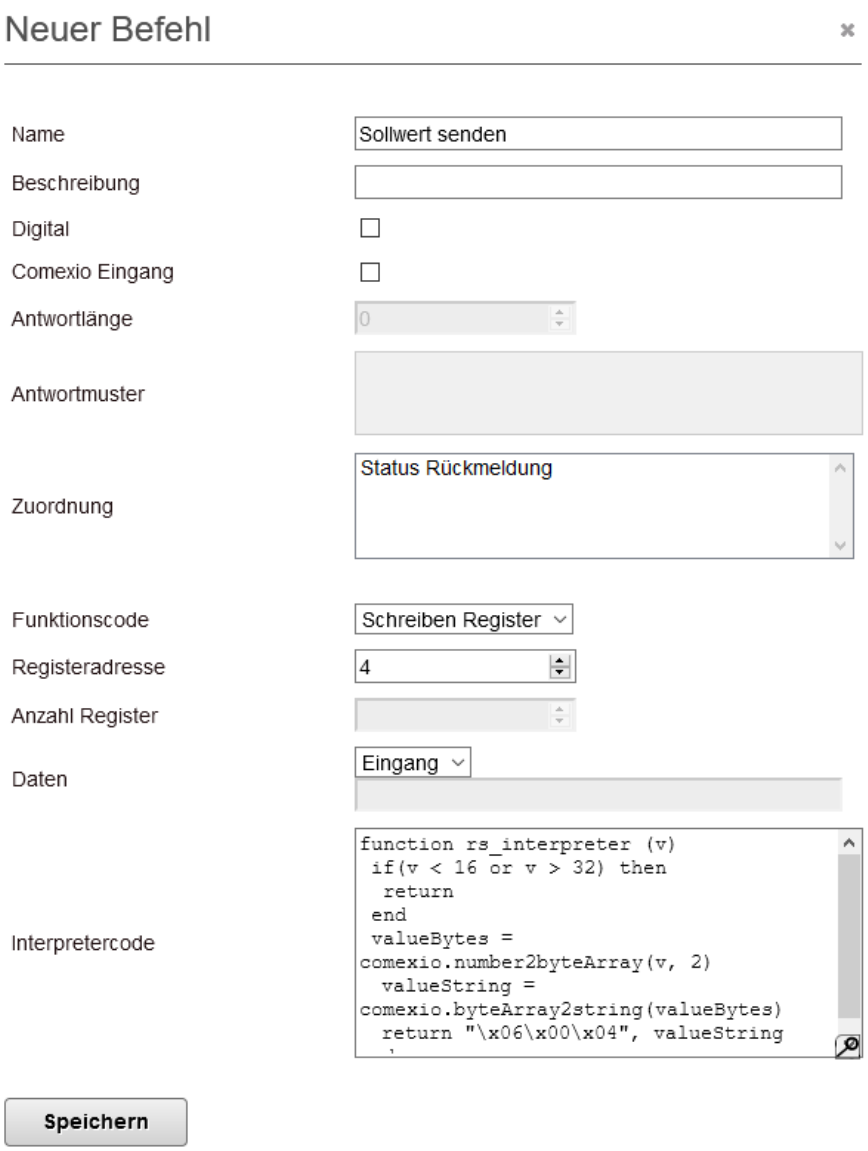
valueBytes = comexio.tableChunk(answerBytes, 4, 5)

Das Array wird auf die relevanten Stellen reduziert. In Stelle 4 und 5 des Array steht der Rückgabewert der gesendeten Nachricht.

return comexio.byteArray2dec(valueBytes)

Das reduzierte Array wird in eine Dezimalzahl umgewandelt und ausgegeben.

Beispiel 3: Sollwert senden




```
function rs_interpreter (v)
  if(v < 16 or v > 32) then
    return
  end
    valueBytes = comexio.number2byteArray(v, 2)
    valueString = comexio.byteArray2string(valueBytes)
    return "\x06\x00\x04", valueString
end
```

Mit folgenden Zeilen wurde eingerichtet, dass nur Werte zwischen 16 und 32 gesendet werden können (siehe Ausschnitt der Anleitung). Werte außerhalb dieses Bereichs werden nicht gesendet.

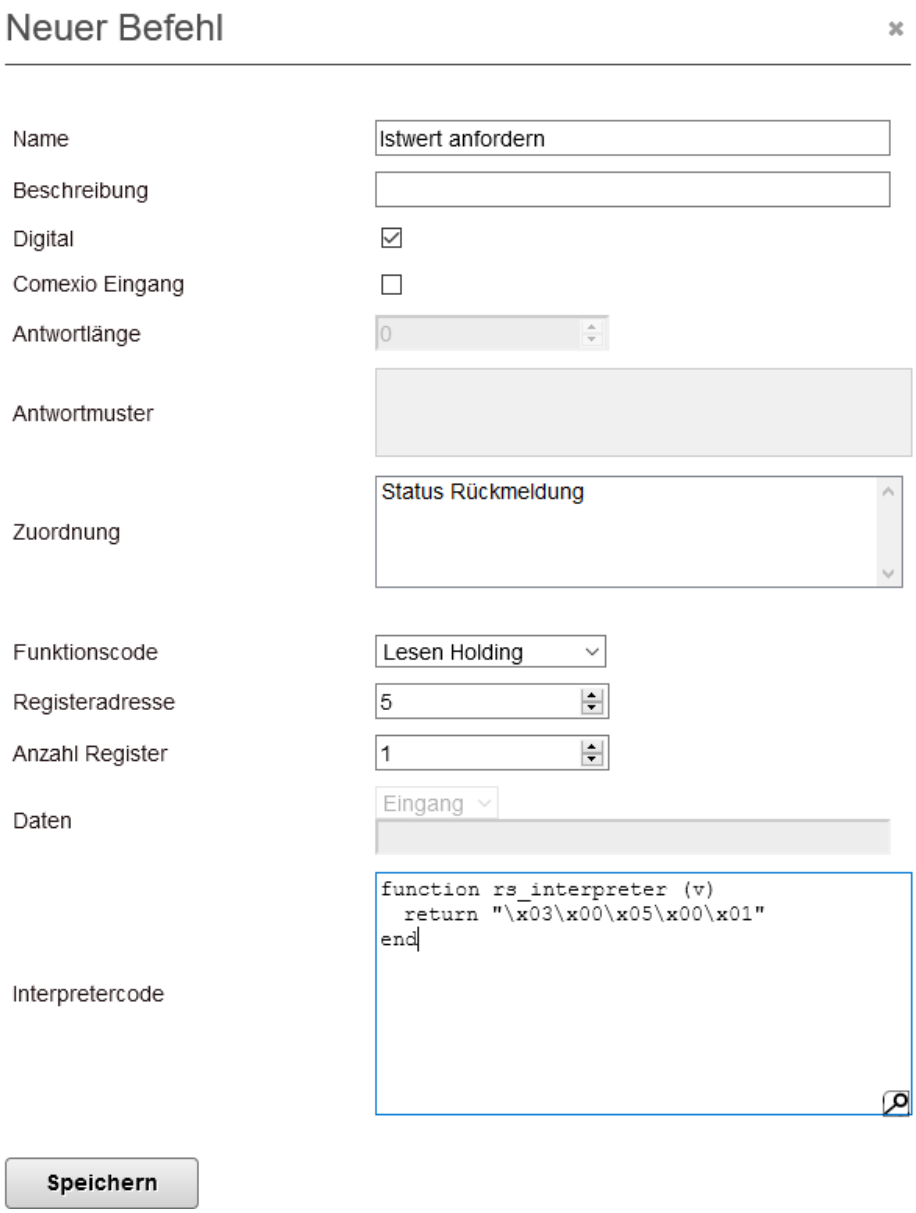
if(v < 16 or v > 32) then

return

end

Beispiel 4: Istwert lesen

Befehl 1: Istwert anfordern



Befehl 2: Istwert interpretieren