# Modbus Slave TCP/IP

## Implementation of Modbus Slave TCP/IP for Alfen NG9xx platform

Author: T. Nederlof

Version: 2.3
30-10-2020

ALFEN
POWER TO ADAPT

# Table of contents

Page

# 1 Introduction

This is a draft of the implementation of the Modbus slave functionality to the NG9xx charging stations of Alfen N.V. All information in this document might be subject to change in the future and should be used as an indication of which functionality will be supported.

There is an important difference between the master and slave roles in the Modbus communication protocol. This implementation covers the slave role that supports serving a Modbus master, these roles are also referred to server and client roles respectively when using Modbus over TCP/IP. The master initiates the connection to the slave and sends either read or write requests of certain Modbus registers.

The current Modbus implementation supports up to two simultaneously connected Modbus TCP/IP masters, UDP is not supported. There is a keep alive timeout of 60 seconds before the connection with a Modbus master is closed when no new read or write is received. The Modbus master should connect to the IP of the Modbus slave's wired Ethernet connection on port 502. Requests with certain slave addresses are accepted, where charging station related Modbus registers require slave address 200 and socket related Modbus registers require slave address 1 or 2, depending on the socket. All communication must be in the big endian format.
New values of received written registers by Modbus masters are logged in the charging station, only if this value is different than the current one.

This document is based on using:

- Firmware version 4.10
- Service Installer Application 3.4.10-130

## 1.1 Configuration options

Configuring the Modbus slave functionality can be done using the Alfen ACE Service Installer[1] with an administrator account or version 4.0 or higher with a service account. The charging station requires the license key for “Active load balancing” before the Modbus slave functionality can be enabled. The following configured settings are persistent and remain preserved when the charging station reboots.

| Name | Function |
|---|---|
| **Allow reading** | Allow reading of Modbus registers via TCP/IP. By default this is turned off. |
| **Allow writing maximum currents** | Allow writing maximum current Modbus registers. By default this is turned off. |
| **Enable sockets** | This enables the charging station to take the written maximum current values for sockets into account when calculating the actual maximum current for all the sockets. |
| **Enable SCN** | This enables the charging station to take the written maximum current values for SCN into account when calculating the actual maximum current for all the SCN phases. |
| **Validity time** | The validity time is the time in seconds in which the charging station requires an updated maximum current from a Modbus master before falling back to the safe current. The validity time is equal for all maximum currents, however each maximum current has its own remaining valid time which is updated every time that maximum current is set via Modbus. The default validity time is 60 seconds. It is recommended that the polling time of a Modbus master is lower than the validity time. |
| **IP Address allocation** | DHCP or fixed IP |
| **Port** | 502 |
| **Modbus slave addresses** | 1: measurements socket 1<br>2: measurements socket 2 (if available) |
| **Supported Modbus functions** | 0x03: Read Holding Registers<br>0x06: Write Single Register<br>0x10: Write Multiple Registers |

## 1.2 Reading registers

The Modbus slave implementation supports reading of holding registers with Modbus function code 3. Multiple registers can be requested within one Modbus request, as long as they are contiguous. Whenever a register is reserved or not available, the register reply is filled with Not a Number (NaN), which is set to 0xFFFF for a 16 bit register.
There are a Modbus registers that contain the datatype string. String registers contain strings where each 16-bit Modbus register contains two 8-bit ASCII chars. A string always contains a trailing zero.

**Note: Reading registers is done in network byte order.**

[1] Available at https://alfen.com/downloads

## 1.3 Writing registers

The Modbus slave implementation supports writing of holding registers. When writing a value with a datatype that contains multiple Modbus registers, then all registers should be written within one write request. For example when a variable is a 32 bit float, both consecutive 16 bit registers should be written in one Modbus request. When the Modbus write request does not write all registers, the request will be denied and a Modbus error will be returned.

**Note: Writing registers is done in network byte order.**

### 1.3.1 Maximum current

It is possible to set the maximum current via Modbus for a specific socket, or for a specific phase of the SCN network. Each maximum current has additional registers that are read only for the enabled property, actual maximum current, the configured safe current and the remaining validity time. Each time the maximum current is written via Modbus, the remaining valid time is updated with the validity time. For example, when the validity time is 60 seconds and the maximum current has been written most recently 10 seconds ago, then reading of the remaining valid time register will result in 50.
The maximum current and remaining valid time are not preserved during a reboot of the charging station. Since the enabled and safe current settings are persistent, the behaviour of the charging station will be such that it will first fall back to the safe current and waits for the Modbus master to rewrite the maximum current.

When a maximum current is enabled and the Modbus master did not update this for a certain time, the charging station will fall back to its safe current. The safe current has to be set before the maximum current can be set via Modbus. This value can also be configured by the back office.
Then internal handling of a newly written maximum current by the charging station can require some time. The time before the actual used current by the connected car is adapted to this relies on multiple factors, for starters by the response speed of the car itself.

## 1.4 Phase rotation

It must be noted that the naming of the phases depend on the installation and thus are for the charging station relative to the incoming phase rotation. This could potential result in confusion on the Modbus master side, especially when multiple Modbus slave charging stations are connected.
An example could be two charging stations of which the first is connected in order of the incoming phase connection L1-L2-L3, while the second charging station is wired in order L2-L3-L1. This could mean that if the Modbus master wants to read the current the current through phase L1, it should read the register of the phase L1 on the first charging station and the register of the phase L2 of the second charging station.

# 2 Enabling Modbus Slave in charging stations

Modbus Slave over TCP/IP is activated when the station is configured to communicate with an Energy Management System (EMS) and when Active Load Balancing is activated. Active Load Balancing is the functionality the currently implements Modbus Slave TCP/IP as a means of communication.

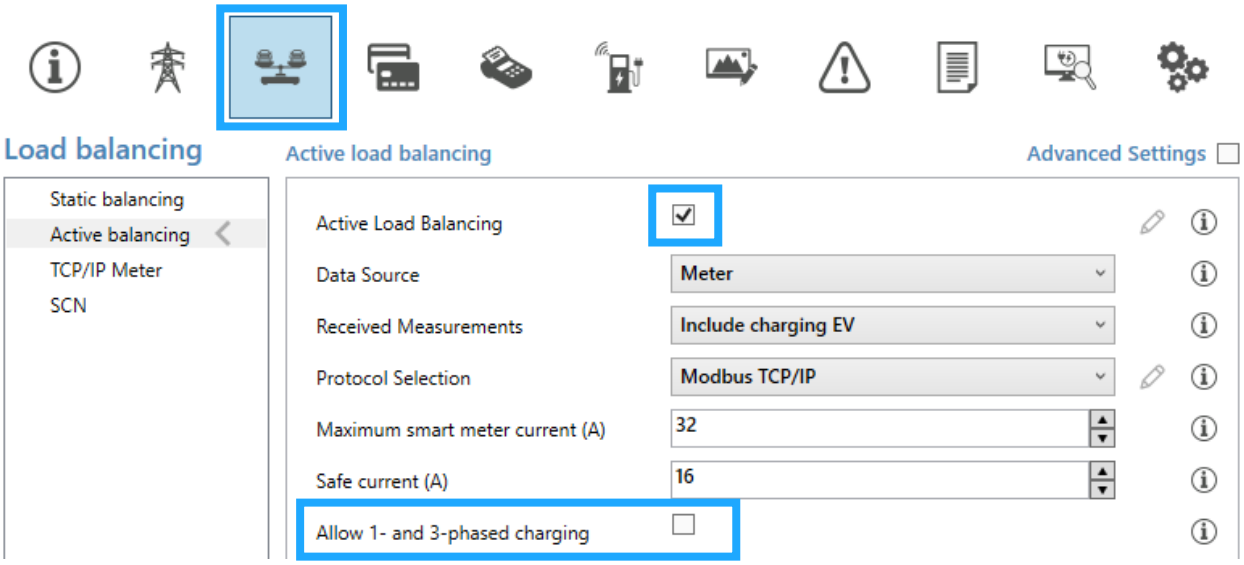
Note: Active Load Balancing is a locked feature of the charging station. It can only be unlocked after purchasing that feature. After the purchase, the unique license key of the charging station is updated.

This chapter displays the steps to take to enable Modbus Slave TCP/IP for e.g. testing purposes.

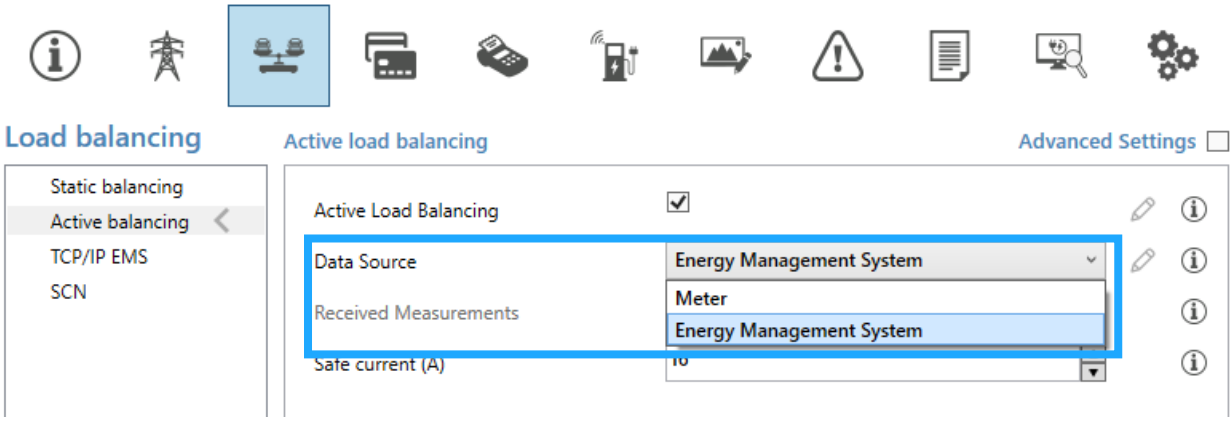
To configure the charging station on site, an account for Service Installer Application (SIA) is required. In case you do not have an account yet, please visit https://support.alfen.com to request an account.

## 2.1 Activate Active Load Balancing

Active Load Balancing can be activated using the selection box in the Section 'Load Balancing' of the Service Installer Application. In the menu on the left, select "Active balancing".




**Note:** the checkbox for 'Allow 1- and 3-phased charging'. This option must be active to allow an Energy Management System to control switching between single-phase and three-phases while charging. In the current implementation this option must be checked locally using the Service Installer Application.




In the Active Balancing menu, select your Data Source:

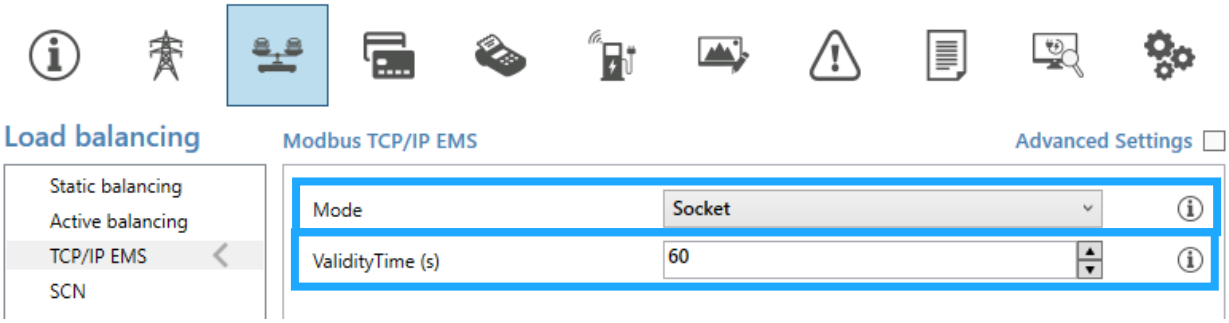
- **Meter**; This will activate ‘Master’ role for the charging station
- **Energy Management System**; This will activate ‘Slave’ role for the charging station

As a ‘Master’, the charging station calculates the available budget left for charging vehicles. Other consumers are considered of higher priority. Charging ramps up only if possible

As a ‘Slave’, the charging station responds to a command from an external device like an Energy Management System (EMS). The external device determines the priority of charging and acts accordingly. The charging station ramps up/down upon request.

## 2.2 Energy Management Systems

When Selecting ‘EMS’, the charging station will be configured as a ‘Slave’. In the menu bar on the left, ‘TCP/IP EMS’ appears. Modbus TCP/IP is selected by default, no other protocol can be selected yet.




The following options are available:

**Mode;**
- **Socket;** control each socket
- **SCN;** control the charging station as a complete station, or control a complete Smart Charging Network as one entity.

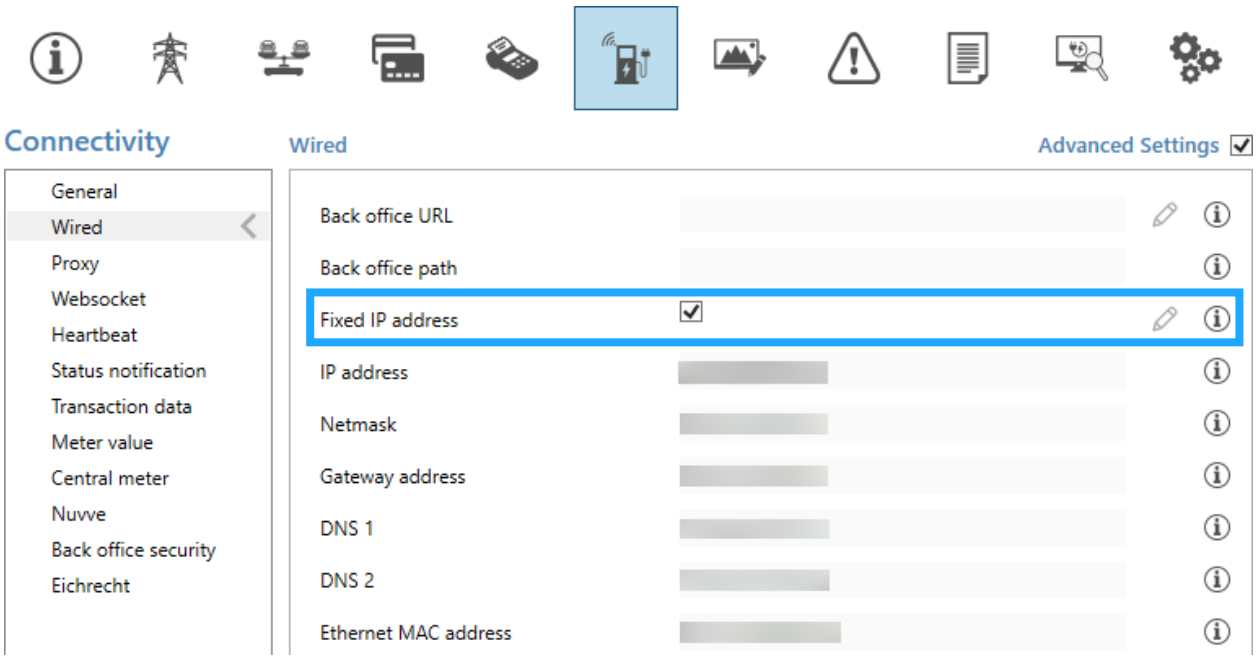
**Validity time (s);** After validity time, the station assumes that the Energy Management System (EMS) is not available anymore and reverts to safe current, configured in menu ‘Active balancing’. Register values must be rewritten before validity time has expired.

**Configuring IP-address of the charging station**
The charging station is by default configured to work with automatic IP allocation using DHCP. This can be used for Modbus Slave TCP/IP operation as well (station being the TCP server). To retrieve the station’s identity use (for example) mDNS to retrieve its DNS name

- **Service type:** _alfen._tcp.local
- **Assigned IP address**
- **Port**: 80
- **Modelname-serial number:** NG920-61001-ACE0012345 (example)
- **Station Identifier** ALF_123 (example)
- **SCN name** (if applicable): ALF_HBW (example)

As an alternative it is also possible to use a fixed IP address on the network to find the station.
- Navigate to tab ‘Connectivity’
- Click ‘Wired’ in the menu on the left
- Select Fixed IP address
- Fill in the details

See figure below for the screen to configure a fixed IP address.


Connectivity
Wired
Advanced Settings
General
Wired
Proxy
Websocket
Heartbeat
Status notification
Transaction data
Meter value
Central meter
Nuvve
Back office security
Eichrecht
Back office URL
Back office path
Fixed IP address
IP address
Netmask
Gateway address
DNS 1
DNS 2
Ethernet MAC address

# 3 Modbus Register table

## 3.1 Product identification registers

The product identification registers can be reached using slave address 200.

| Description | Start address | End address | Number of 16 bit registers | Read or Write | Data Type | Step size & Units | Additional info |
|---|---|---|---|---|---|---|---|
| **Name** | 100 | 116 | 17 | R | STRING | n.a. | "ALF_1000" |
| **Manufacturer** | 117 | 121 | 5 | R | STRING | n.a. | "Alfen NV" |
| **Modbus table version** | 122 | 122 | 1 | R | SIGNED16 | n.a. | 1 |
| **Firmware version** | 123 | 139 | 17 | R | STRING | n.a. | "3.4.0-2990" |
| **Platform type** | 140 | 156 | 17 | R | STRING | n.a. | "NG910" |
| **Station serial number** | 157 | 167 | 11 | R | STRING | n.a. | "00000R000" |
| **Date year** | 168 | 168 | 1 | R | SIGNED16 | 1yr | 2019 |
| **Date month** | 169 | 169 | 1 | R | SIGNED16 | 1mon | 03 |
| **Date day** | 170 | 170 | 1 | R | SIGNED16 | 1d | 11 |
| **Time hour** | 171 | 171 | 1 | R | SIGNED16 | 1hr | 12 |
| **Time minute** | 172 | 172 | 1 | R | SIGNED16 | 1min | 01 |
| **Time second** | 173 | 173 | 1 | R | SIGNED16 | 1s | 04 |
| **Uptime** | 174 | 177 | 4 | R | UNSIGNED64 | 0.001s | 100 |
| **Time zone** | 178 | 178 | 1 | R | SIGNED16 | 1min | Time zone offset to UTC in minutes |

## 3.2 Station status registers

The station status registers can be reached using slave address 200.

| Description | Start address | End address | Number of 16 bit registers | Read or Write | Data Type | step size & Units | Additional info |
|---|---|---|---|---|---|---|---|
| **Station Active Max Current** | 1100 | 1101 | 2 | R | FLOAT32 | 1A | The actual max current |
| **Temperature** | 1102 | 1103 | 2 | R | FLOAT32 | 1°C | Board temperature, does not reflect environment temperature |
| **OCPP state** | 1104 | 1104 | 1 | R | UNSIGNED16 | N.A. | To verify whether back office is connected |
| **Nr of sockets** | 1105 | 1105 | 1 | R | UNSIGNED16 | N.A. | Number of sockets |

## 3.3 SCN registers

The SCN registers can be reached using slave address 200.

| Description | Start address | End address | Number of 16 bit registers | Read or Write | Data Type | step size & Units | Additional info |
|---|---|---|---|---|---|---|---|
| **SCN name** | 1400 | 1403 | 4 | R | STRING | n.a. | |
| **SCN Sockets** | 1404 | 1404 | 1 | R | UNSIGNED16 | 1A | Number of configured sockets |
| **SCN Total Consumption Phase L1** | 1405 | 1406 | 2 | R | FLOAT32 | 1A | |
| **SCN Total Consumption Phase L2** | 1407 | 1408 | 2 | R | FLOAT32 | 1A | |
| **SCN Total Consumption Phase L3** | 1409 | 1410 | 2 | R | FLOAT32 | 1A | |
| **SCN Actual Max Current Phase L1** | 1411 | 1412 | 2 | R | FLOAT32 | 1A | |
| **SCN Actual Max Current Phase L2** | 1413 | 1414 | 2 | R | FLOAT32 | 1A | |
| **SCN Actual Max Current Phase L3** | 1415 | 1416 | 2 | R | FLOAT32 | 1A | |
| **SCN Max Current per Phase L1** | 1417 | 1418 | 2 | R/W | FLOAT32 | 1A | |
| **SCN Max Current per Phase L2** | 1419 | 1420 | 2 | R/W | FLOAT32 | 1A | |
| **SCN Max Current per Phase L3** | 1421 | 1422 | 2 | R/W | FLOAT32 | 1A | |
| **Remaining valid time Max Current Phase L1** | 1423 | 1424 | 2 | R | UNSIGNED32 | 1s | Max current valid time |
| **Remaining valid time Max Current Phase L2** | 1425 | 1426 | 2 | R | UNSIGNED32 | 1s | Max current valid time |
| **Remaining valid time Max Current Phase L3** | 1427 | 1428 | 2 | R | UNSIGNED32 | 1s | Max current valid time |
| **SCN Safe current** | 1429 | 1430 | 2 | R | FLOAT32 | 1A | Configured SCN safe current |
| **SCN Modbus Slave Max Current enable** | 1431 | 1431 | 1 | R | UNSIGNED16 | n.a. | 1:Enabled, 0: Disabled. |

## 3.4 Socket measurement registers

The socket measurements show information regarding the energy meter that is connected to the only socket in case of a single socket charging station, or the left socket in case of a dual socket charging station and can be reached using slave address 1. In case of a dual socket station, the right socket related energy measurements can be reached using slave address 2.

| Description | Start address | End address | Number of 16 bit registers | Read or Write | Data Type | Step size & Units | Additional info |
|---|---|---|---|---|---|---|---|
| **Energy measurements** | | | | | | | |
| **Meter state** | 300 | 300 | 1 | R | UNSIGNED16 | n.a. | Bitmask with state:<br>Initialised: 0x01<br>Updated: 0x02<br>Warning: 0x04<br>Error: 0x08 |
| **Meter last value timestamp** | 301 | 304 | 4 | R | UNSIGNED64 | 0.001s | Milliseconds since last received measurement |
| **Meter type** | 305 | 305 | 1 | R | UNSIGNED16 | n.a. | 0:RTU, 1:TCP/IP, 2:UDP, 3:P1, 4:other |
| **Voltage Phase V(L1-N)** | 306 | 307 | 2 | R | FLOAT32 | 1V | |
| **Voltage Phase V(L2-N)** | 308 | 309 | 2 | R | FLOAT32 | 1V | |
| **Voltage Phase V(L3-N)** | 310 | 311 | 2 | R | FLOAT32 | 1V | |
| **Voltage Phase V(L1-L2)** | 312 | 313 | 2 | R | FLOAT32 | 1V | |
| **Voltage Phase V(L2-L3)** | 314 | 315 | 2 | R | FLOAT32 | 1V | |
| **Voltage Phase V(L3-L1)** | 316 | 317 | 2 | R | FLOAT32 | 1V | |
| **Current N** | 318 | 319 | 2 | R | FLOAT32 | 1A | |
| **Current Phase L1** | 320 | 321 | 2 | R | FLOAT32 | 1A | |
| **Current Phase L2** | 322 | 323 | 2 | R | FLOAT32 | 1A | |
| **Current Phase L3** | 324 | 325 | 2 | R | FLOAT32 | 1A | |
| **Current Sum** | 326 | 327 | 2 | R | FLOAT32 | 1A | |
| **Power Factor Phase L1** | 328 | 329 | 2 | R | FLOAT32 | N.A. | |
| **Power Factor Phase L2** | 330 | 331 | 2 | R | FLOAT32 | N.A. | |
| **Power Factor Phase L3** | 332 | 333 | 2 | R | FLOAT32 | N.A. | |
| **Power Factor Sum** | 334 | 335 | 2 | R | FLOAT32 | N.A. | |
| **Frequency** | 336 | 337 | 2 | R | FLOAT32 | 1Hz | |
| **Real Power Phase L1** | 338 | 339 | 2 | R | FLOAT32 | 1W | |
| **Real Power Phase L2** | 340 | 341 | 2 | R | FLOAT32 | 1W | |
| **Real Power Phase L3** | 342 | 343 | 2 | R | FLOAT32 | 1W | |
| **Real Power Sum** | 344 | 345 | 2 | R | FLOAT32 | 1W | |
| **Apparent Power Phase L1** | 346 | 347 | 2 | R | FLOAT32 | 1VA | |
| **Apparent Power Phase L2** | 348 | 349 | 2 | R | FLOAT32 | 1VA | |
| **Apparent Power Phase L3** | 350 | 351 | 2 | R | FLOAT32 | 1VA | |
| **Apparent Power Sum** | 352 | 353 | 2 | R | FLOAT32 | 1VA | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Reactive Power Phase L1** | 354 | 355 | 2 | R | FLOAT32 | 1VAr | |
| **Reactive Power Phase L2** | 356 | 357 | 2 | R | FLOAT32 | 1VAr | |
| **Reactive Power Phase L3** | 358 | 359 | 2 | R | FLOAT32 | 1VAr | |
| **Reactive Power Sum** | 360 | 361 | 2 | R | FLOAT32 | 1VAr | |
| **Real Energy Delivered Phase L1** | 362 | 365 | 4 | R | FLOAT64 | 1Wh | |
| **Real Energy Delivered Phase L2** | 366 | 369 | 4 | R | FLOAT64 | 1Wh | |
| **Real Energy Delivered Phase L3** | 370 | 373 | 4 | R | FLOAT64 | 1Wh | |
| **Real Energy Delivered Sum** | 374 | 377 | 4 | R | FLOAT64 | 1Wh | |
| **Real Energy Consumed Phase L1** | 378 | 381 | 4 | R | FLOAT64 | 1Wh | |
| **Real Energy Consumed Phase L2** | 382 | 385 | 4 | R | FLOAT64 | 1Wh | |
| **Real Energy Consumed Phase L3** | 386 | 389 | 4 | R | FLOAT64 | 1Wh | |
| **Real Energy Consumed Sum** | 390 | 393 | 4 | R | FLOAT64 | 1Wh | |
| **Apparent Energy Phase L1** | 394 | 397 | 4 | R | FLOAT64 | 1VAh | |
| **Apparent Energy Phase L2** | 398 | 401 | 4 | R | FLOAT64 | 1VAh | |
| **Apparent Energy Phase L3** | 402 | 405 | 4 | R | FLOAT64 | 1VAh | |
| **Apparent Energy Sum** | 406 | 409 | 4 | R | FLOAT64 | 1VAh | |
| **Reactive Energy Phase L1** | 410 | 413 | 4 | R | FLOAT64 | 1VArh | |
| **Reactive Energy Phase L2** | 414 | 417 | 4 | R | FLOAT64 | 1VArh | |
| **Reactive Energy Phase L3** | 418 | 421 | 4 | R | FLOAT64 | 1VArh | |
| **Reactive Energy Sum** | 422 | 425 | 4 | R | FLOAT64 | 1VArh | |
| **Status and transaction registers** | | | | | | | |
| **Availability** | 1200 | 1200 | 1 | R | UNSIGNED16 | n.a. | 1: Operative, 0: inoperative |
| **Mode 3 state** | 1201 | 1205 | 5 | R | STRING | n.a. | 61851 states |
| **Actual Applied Max Current** | 1206 | 1207 | 2 | R | FLOAT32 | 1A | Actual Applied overall Max Current for socket |
| **Modbus Slave Max Current valid time** | 1208 | 1209 | 2 | R | UNSIGNED32 | 1s | Remaining time before fall back to safe current |
| **Modbus Slave Max Current** | 1210 | 1211 | 2 | R/W | FLOAT32 | 1A | |
| **Active Load Balancing Safe Current** | 1212 | 1213 | 2 | R | FLOAT32 | 1A | Active Load Balancing safe current |
| **Modbus Slave received setpoint accounted for** | 1214 | 1214 | 1 | R | UNSIGNED16 | n.a. | 1:Yes, 0: No |
| **Charge using 1 or 3 phases** | 1215 | 1215 | 1 | R/W | UNSIGNED16 | phases | 1: 1 phase, 3: 3 phase charging |

**Note:** Register 1214 *'Modbus Slave received setpoint accounted for'* indicates whether the received Max Current (registers 1210-1211), also called a setpoint, is taken into account to determine the Actual

Applied Max Current (registers 1206-1207). Depending on other setpoints, registers 1206-1207 may read the settings requested by the external device (e.g. EMS).

### 3.4.1 Mode 3 state listing

| State | Signal voltage (DC) | PWM signal applied | Vehicle connected | Charging |
|---|---|---|---|---|
| **A** | 12V | No | No | No |
| **B1** | 9V | No | Yes | No |
| **B2** | 9V | Yes | Yes | No |
| **C1** | 6V | No | Yes | No |
| **C2** | 6V | Yes | Yes | Yes |
| **D1** | 3V | No | Yes | No |
| **D2** | 3V | Yes | Yes | Yes |
| **E** | 0V | No | No | No |
| **F** | -12V | No | No | No |

Note that State F is an error state.