MENNEKES
MY POWER CONNECTION

ECU-BRx and ECU-BBx
Modbus TPC Server Specification

Doc. Revsion: 1.07
Date: Jun. 10, 22

Table of contents

1 Release information

The Modbus TCP Server interface which is described in this document will be available from the software version ≥ 5.12.x for our ECU, used in AMEDIO Professional, AMTRON Professional and AMTRON Charge Control.

Note that starting from version 5.22, the implementation includes two additional 32-bit registers which provide improved functionality for CHARGE_DURATION and CHARGED_ENERGY, described also in this document.

2 Configuration parameters available on the Charge-Controller's user interface

Parameter Name	Description
Modbus TCP Server	Allows to turn the Charge Point into a Modbus TCP Server. This allows reading and writing parameters using the Modbus protocol. See the documentation for detailed register information.
Modbus TCP Server Base Port	Port number on which the Modbus TCP Server waits for incoming connections on connector 1. In case a second connector is supported, the configured 'port + 1' will be used for that connector.
Modbus TCP Server Register Address Set	Choose the set of register addresses that the Modbus TCP Server device will expose to its client.
Modbus TCP Server Allow Start/Stop Transaction	Allows transactions to be started/stopped from a Modbus Master device via the controller's Modbus TCP Server interface.
Modbus TCP Server Allow UID Disclose	Allows sending the UID over the Modbus TCP Server protocol

3 General System Information

Modbus Unit ID:

The Modbus TCP Server on the charge controller will reply to messages with any Unit ID from 1 to 255.

Modbus TCP Server Base Port +1:

If you want to access a charging station with more than one connector, you need to connect to each Charge-Controller dedicated to one connector.

To connect to each Charge-Controller separately on the AMEDIO Pole with two charging points you have to use two separate ports. For example: when the master Charge-Controller is configured on the 502 port (default configuration), the slave Charge-Controller had always the port number one bigger than the master. In this example the slave Charge-Controller had the port number 503.

Lowering charging current:

To lower the charging current the HEMS shall write to the Register HEMS_CURRENT_LIMIT as described in section “HEMS configuration options”.

Please note the actual signalled current as indicated by register 706 can be lower than the HEMS_CURRENT_LIMIT since other limitations (such as charging cable or Dynamic Load Management limits) could apply.

Please note the actual signalled current as indicated by the register named SIGNALED_CURRENT can be lower than the HEMS_CURRENT_LIMIT since other limitations (such as charging cable, or dynamic load management limits) could apply.

Word and byte ordering:

With the notable exception of the values in the register destined to error codes (those with names prefixed with ERROR_CODES_) which are explained separately in their corresponding section, all other registers are to be read and written with the high byte first and the low byte after. For double registers (32-bit) the order of the words is the high word first and the low word after.

As an example, if registers 200-201 are read and contain a value of 0x0001 for register 200 and a value of 0x1F40 for register 201, these values are to be read as 0x00011F40 that is a decimal value of 73536.

General system information:

The following register section contain the general system Information:

The registers in the first table that follows contain general information about the system, about its status, error states, FW and protocol versions and other system and configuration information.

For details on how to read and interpret the registers designated for error handling please refer to the section Error states mask mappings.

Register Addr.	Bit field	Size	Type	Name	Description
100-101	31:0	32-Bit	READ	FIRMWARE_VERSION	ECU Application version number (example: 0.91 = {0x30, 0x2E, 0x39, 0x31) 4.40 = {0x34, 0x2E, 0x34, 0x34}).
104	15:0	16-Bit	READ	OCPP_CP_STATUS	0 = Available 1 = Occupied 2 = Reserved 3 = Unavailable 4 = Faulted 5 = Preparing 6 = Charging 7 = Suspend-edEVSE 8 = SuspendedEV 9 = Finishing
105-106	31:0	32-Bit	READ	ERROR_CODES_1	See **Error States Mask Mappings** section for mask bit values.
107-108	63:32	32-Bit	READ	ERROR_CODES_2	
109-110	95:64	32-Bit	READ	ERROR_CODES_3	
111-112	127:96	32-Bit	READ	ERROR_CODES_4	
120-121	31:0	32-Bit	READ	PROTOCOL_VERSION	ECU Modbus Protocol Version number (example: 0.6 = {0x30, 0x2E, 0x36}).

					Current Version 0.9c
122	15:0	16-Bit	READ	VEHICLE_STATE	Control Pilot vehicle state in decimal format: A = 1, B = 2, C = 3, D = 4, E = 5
124	15:0	16-Bit	R/W	CP_AVAILABILITY	Read or set the Charge Point availability. Software version < 5.22: 0 = available, 1 = unavailable Software version ≥ 5.22 0 = unavailable 1 = available
130	15:0	16-Bit	R/W	MODBUS_ADDRESS_OFFSET	Indicates an offset for the register numbers specified in this document. This register has the value 0 by default.
131	15:0	16-Bit	R/W	SAFE_CURRENT	Max. charge current under communication failure with the Modbus Master.
132	15:0	16-Bit	R/W	COMM_TIMEOUT	Communication timeout with Modbus Master after

					which the Slave will fall back to the current limit specified in the SAFE_CURRENT register.
134	15:0	16-Bit	READ	OPERATOR_CURRENT_LIMIT	Operator Current limit in Amps Software version ≥ 5.22
142-143	31:0	32-Bit	READ	CHARGEPOINT_MODEL_1	Identifies the model of the Charge Point as set by the manufacturer Value starting with • ICC1622 -> AMEDIO • CC612 -> AMTRON Prof. • _CC -> AMTRON Charge Control Software version ≥ 5.22
144-145	31:0	32-Bit	READ	CHARGEPOINT_MODEL_2	
146-147	31:0	32-Bit	READ	CHARGEPOINT_MODEL_3	
148-149	31:0	32-Bit	READ	CHARGEPOINT_MODEL_4	
150-151	31:0	32-Bit	READ	CHARGEPOINT_MODEL_5	
152	15:0	16-Bit	READ	PLUG_LOCK_STATUS	Status of plug lock detection: 1-Locked, 0- Unlocked. Software version ≥ 5.22

A note on OCPP_CP_STATUS: The Charge Point Status as defined in OCPP 1.5 and 1.6 is available in the OCPP_CP_STATUS register. Note that there is some difference between the two versions, so for example the value 1, which is used for "Occupied", is only valid for OCPP 1.5 and instead in 1.6 a more detailed value is available. For the descriptions of each of the charge point status values, please refer to the OCPP specification that corresponds to the version in use.

4 Meter values from OCPP primary Meter

Meter values are unsigned and sent in 32-bit words.

When not available, values will be initialized with a **0xffffffff** value to indicate that no meter is present.

On installations where line-specific Power and Energy are not available, either because the installation is single-phase, or because the meter does not support those readings, only the Total Power and Total Energy will be available.

For maintaining backwards compatibility with previous systems, where the METER_TOTAL_ENERG and METER_TOTAL_POW registers were not present, the Total Power and Total Energy values can also be read from the Power and Energy registers corresponding to L1. In that case the Power and Energy registers for L2 and L3 will return 0xffffffff.

Register Addr.	Bit field	Size	Type	Name	Description
200-201	31:0	32-Bit	READ	METER_ENERG_L1	Energy in Wh. (phase 1) from primary meter
202-203	31:0	32-Bit	READ	METER_ENERG_L2	Energy in Wh. (phase 2) from primary Meter
204-205	31:0	32-Bit	READ	METER_ENERG_L3	Energy in Wh. (phase 3) from primary meter
206-207	31:0	32-Bit	READ	METER_POW_L1	Power in W (phase 1) from primary meter
208-209	31:0	32-Bit	READ	METER_POW_L2	Power in W (phase 2) from primary meter
210-211	31:0	32-Bit	READ	METER_POW_L3	Power in W (phase 3) from primary meter
212-213	31:0	32-Bit	READ	METER_CUR_L1	Current in mA (phase 1) from primary meter
214-215	31:0	32-Bit	READ	METER_CUR_L2	Current in mA (phase 2) from primary meter
216-217	31:0	32-Bit	READ	METER_CUR_L3	Current in mA (phase 3) from primary meter
218-219	31:0	32-Bit	READ	METER_TOTAL_ENERG	Total Energy in Wh. from primary meter

					Software version ≥ 5.22
220-221	31:0	32-Bit	READ	METER_TOTAL_POW	Total Power in W from primary meter Software version ≥ 5.22
222-223	31:0	32-Bit	READ	METER_VOL_L1	Voltage in V. (phase 1) from primary meter Software version ≥ 5.22
224-225	31:0	32-Bit	READ	METER_VOL_L2	Voltage in V. (phase 2) from primary meter Software version ≥ 5.22
226-227	31:0	32-Bit	READ	METER_VOL_L3	Voltage in V. (phase 3) from primary meter Software version ≥ 5.22

Support of the above registers and corresponding values are dependent on the specific meter that is built into the product. The following table gives further details on the combinations used and the corresponding registers:

Address	**Name (short)**	**Meter**				
		ABB EV3	**EMH eHZ**	**GMZ EM2289**	**NZR S85**	**NZR SL85**
200	Energy L1	Total Energy	Total Energy	Total Energy	Total Energy	Total Energy
202	Energy L2	no	no	no	no	no
204	Energy L3	no	no	no	no	no
206	Power L1	yes	yes	yes	Total Power	yes
208	Power L2	yes	yes	yes	no	yes
210	Power L3	yes	yes	yes	no	yes
212	Current L1	yes	yes	yes	yes	yes
214	Current L2	yes	yes	yes	yes	yes
216	Current L3	yes	yes	yes	yes	yes
218	Total Energy	yes	yes	yes	yes	yes
220	Total Power	yes	yes	yes	yes	yes

222	Voltage L1	yes	no	yes	yes	yes
224	Voltage L2	yes	no	yes	yes	yes
226	Voltage L3	yes	no	yes	yes	yes

5 Dynamic Load Management

This is information mostly concerning the DLM Master. Everything except for register 600 will be available only for devices with a DLM Master role set (meaning a value of 1 or 2 is returned on this register).

Register Addr.	Bit field	Size	Type	Name	Description
600	15:0	16-Bit	READ	DLM_MODE	Indicates the DLM mode configured for this device. The following values represent each mode: 0 = Disabled 1 = DLM Master (With internal DLM-Slave) 2 = DLM Master (Standalone) 3 = DLM Slave (Master-Auto-Discovery) 4 = DLM Slave (Master-Fixed-IP)
610	15:0	16-Bit	READ	DLM_EVSE_SUB_DISTRIBUTION_LIMIT_L1	Overall current limit for DLM available for distribution to EVs. These three registers are for L1, L2, and L3 respectively and the values are in Amps.
611	15:0	16-Bit	READ	DLM_EVSE_SUB_DISTRIBUTION_LIMIT_L2	
612	15:0	16-Bit	READ	DLM_EVSE_SUB_DISTRIBUTION_LIMIT_L3	
613	15:0	16-Bit	R/W	DLM_OPERATOR_EVSE_SUB_DISTRIBUTION_LIMIT_L1	Operator current limit for DLM available for distribution to EVs. The 'Operator EVSE Sub-Distribution Limit' is equal or smaller than the 'EVSE Sub-Distribution Limit'.
614	15:0	16-Bit	R/W	DLM_OPERATOR_EVSE_SUB_DISTRIBUTION_LIMIT_L2	

615	15:0	16-Bit	R/W	DLM_OPERATOR_EVSE_SUB_DISTRIBUTION_LIMIT_L3	These three registers are for L1, L2, and L3 respectively and the values are in Amps.
620	15:0	16-Bit	READ	DLM_EXTERNAL_METER_SUPPORT	If enabled, an external, secondary meter allows to also consider the power consumption of additional load. The power available for charging EVs will be adjusted accordingly. Please make sure, 'Meter configuration (Second)' is configured, preferably to a 3-phase, phase aware meter. Value of this register is 1 when enabled, 0 when disabled.
621	15:0	16-Bit	READ	DLM_NUM_SLAVES_CONNECTED	The number of DLM Slaves connected to this Master device.
630	15:0	16-Bit	READ	DLM_OVERALL_CURRENT_APPLIED_L1	Overall Current the DLM Master is currently applying (sum of current distributed among the slaves). These three registers are for L1, L2, and L3 respectively and the values are in Amps.
631	15:0	16-Bit	READ	DLM_OVERALL_CURRENT_APPLIED_L2	
632	15:0	16-Bit	READ	DLM_OVERALL_CURRENT_APPLIED_L3	
633	15:0	16-Bit	READ	DLM_OVERALL_CURRENT_AVAILABLE_L1	Overall Current the DLM Master has available to distribute among the slaves. These three registers are for L1, L2, and L3 respectively and the values are in Amps.
634	15:0	16-Bit	READ	DLM_OVERALL_CURRENT_AVAILABLE_L2	
635	15:0	16-Bit	READ	DLM_OVERALL_CURRENT_AVAILABLE_L3	

6 Charge process information

This is information collected during, or inferred from, the charging process.

A note on registers 716-717 and 718-719: these registers are nothing more but expanded versions of registers 705 and 709 respectively. Being 32-Bit-sized allow for larger values.

In the case of registers 705 and 709, once the values in those registers reach their maximum, they will stay at the maximum value until the next session is started or until the charging process is finished according to each case.

While registers 705 and 709 will still be supported in future releases, please consider implementing support for 716-717 and 718-719 when possible for improved functionality.

Register Addr.	Bit field	Size	Type	Name	Description
700	15:0	16-Bit	READ	REQ_ENERGY_15118	EV Required Energy – 15118 only
701-702	31:0	32-Bit	READ	SCHED_DEP_TIME_15118	Scheduled departure time (format is `hhmmss` in big-endian packed BCD with left zero padding) – 15118 only
703-704	31:0	32-Bit	READ	SCHED_DEP_DATE_15118	Scheduled departure time (format is `ddmmyy` in big-endian packed BCD with left zero padding) – 15118 only
705	15:0	16-Bit	READ	CHARGED_ENERGY	Sum of charged energy for the current session (Wh). The amount of charged energy will stay at its maximum until the next session is started
706	15:0	16-Bit	READ	SIGNALED_CURRENT	The maximum current that's being signalled to the EV for charging
707-708	31:0	32-Bit	READ	START_TIME	Format is the same as in SCHED_DEP_TIME_15118
709	15:0	16-Bit	READ	CHARGE_DURATION	Duration of the charging process in seconds

710-711	31:0	32-Bit	READ	END_TIME	Format is the same as in S CHED_DEP_TIME_15118
712	15:0	16-Bit	READ	MINIMUM_CUR_LIMIT	Minimum current limit for charging
713-714	31:0	32-Bit	READ	REQ_ENERGY_15118	EV Required Energy – 15118 only
715	15:0	16-Bit	READ	MAX_CUR_EV	This is the maximum current with which the EV can charge. It may come from the cable's PR value in analog vehicles, or from the maximum reported by the vehicle via 15118, whichever is lower. Software version ≥ 5.22
716-717	31:0	32-Bit	READ	CHARGED_ENERGY (SW version ≥ 5.22)	Sum of charged energy for the current session (Wh). The amount of charged energy will stay at its maximum until the next session is started Identical as register 705, ex-cept that it allows for larger (32-bit) value. Software version ≥ 5.22
718-719	31:0	32-Bit	READ	CHARGE_DURATION (SW version ≥ 5.22)	Duration of the charging process in seconds Identical as register 709, ex-cept that it allows for a larger (32-Bit) value Software version ≥ 5.22

<table>
<tr><td>720-721</td><td>31:0</td><td>32-Bit</td><td>READ</td><td>IDTAG_1</td><td rowspan="5">OCPP IdTag. This is a non-null terminated string with a max. length of 20 bytes, represented here in five 32-bit registers (or ten consecutive 16-bit regs.).

The string is padded with blank-space characters on the left, or completely filled with blank-space characters when no IdTag is present.</td></tr>
<tr><td>722-723</td><td>31:0</td><td>32-Bit</td><td>READ</td><td>IDTAG_2</td></tr>
<tr><td>724-725</td><td>31:0</td><td>32-Bit</td><td>READ</td><td>IDTAG_3</td></tr>
<tr><td>726-727</td><td>31:0</td><td>32-Bit</td><td>READ</td><td>IDTAG_4</td></tr>
<tr><td>728-729</td><td>31:0</td><td>32-Bit</td><td>READ</td><td>IDTAG_5</td></tr>
<tr><td>741-742</td><td>31:0</td><td>32-Bit</td><td>READ</td><td>EVCCID_15118_1</td><td rowspan="3">ASCII representation of Hex. Values corresponding to the EVCCID. The EVCCID value is 6 bytes, so the ASCII Hex. representation of it has a length of exactly 12 bytes.

This is a non-null terminated string.

Values are all zero when no vehicle is connected, or when the vehicle is not a 15118-capable smart vehicle.</td></tr>
<tr><td>743-744</td><td>31:0</td><td>32-Bit</td><td>READ</td><td>EVCCID_15118_2</td></tr>
<tr><td>745-746</td><td>31:0</td><td>32-Bit</td><td>READ</td><td>EVCCID_15118_3</td></tr>
</table>

7 HEMS configuration options

Register Addr.	Bit field	Size	Type	Name	Description
1000	15:0	16-Bit	R/W	HEMS_CURRENT_LIMIT	Current limit of the HEMS module in Amps. This register is intended to be modified by an Energy Manager. If the charge session shall be paused, the register needs to be set to “0”; Default value changed from 6 to 32 Software version ≥ 5.22

8 Authorization with IDTag

Please note that for these registers to be enabled, the corresponding option must be set in the controller EMS/Modbus setting named Modbus Slave Allow Start/Stop Transaction.

When writing to these registers, the effect will be exactly the same as if one physically presented an RFID card in front of the card reader. That means it will start/stop the transaction accordingly based on each scenario’s workflow.

Note that the registers in this table are WRITE only. To READ the IDTAG currently in use please refer to the registers prefixed with READ_ID_TAG_ on their name.

Register Addr.	Bit field	Size	Type	Name	Description
1110-1111	31:0	32-Bit	WRITE	WRITE_IDTAG_1	Same format as registers 720-729. If the IdTag string is fewer than 20 characters, it must be padded on the left (the lowermost registers) with blank-space ASCII characters.
1112-1113	31:0	32-Bit	WRITE	WRITE_IDTAG_2	
1114-1115	31:0	32-Bit	WRITE	WRITE_IDTAG_3	
1116-1117	31:0	32-Bit	WRITE	WRITE_IDTAG_4	
1118-1119	31:0	32-Bit	WRITE	WRITE_IDTAG_5	

9 Error States Mask Mappings

In order to represent any simultaneous error states, the value read from the ERROR_CODES registers can be AND'ed with different mask mappings to identify which individual errors may be present at any given time in the system.

Since only bits 0 to 21 are used in the current specification, it is possibly to read only from registers 111-112 in order to optimize the fetching of values.

For completion however, it is clarified here how to read and interpret the values when reading all 8 registers.

To test for each error individually, the resulting value of reading all registers has to be masked against the corresponding error mask. An example is given below on how to achieve this.

Supposing the following value (presented here in HEX) is read from the ERROR_CODES registers:

Register:	105-106	107-108	109-110	111-112
Value:	0000 0000	0000 0000	0000 0000	4100 0000

Please focus first in the value of registers 111-112.

The bits of a double word are numbered from 0 through 31 with bit 0 being the least significant bit. The word containing bit 0 is the low word and the word containing bit 31 is the high word.

Each 32-bit register has the low word first, and each word has the low byte first.

So in the case of registers 111-112 the words must first be inverted, to get a value of 0000 4100, and then the bytes of each word must be inverted too, in order to get a value of 0000 0041.

That is to be done for each register pair (105-106, 107-108, 109-110, 111-112), and once it is done, then the whole resulting value can be simply assembled from all registers by starting from a value of 0, and going through each word (register) in order, first shifting to the left then adding. The result from the case above would be:

0000 0000 0000 0000 0000 0000 0000 0041

Finally, to identify which individual errors are present this value must be AND'ed to each error mask. So by doing the following operation:

0000 0000 0000 0000 0000 0000 0000 0041

AND

0000 0000 0000 0000 0000 0000 0000 0001

It is possible to identify that an error "ERR_RCMB_TRIGGERED" is present. And by continuing doing for example:

0000 0000 0000 0000 0000 0000 0000 0041

AND

0000 0000 0000 0000 0000 0000 0000 0040

It can be observed that also there is an error "ERR_CONTACTOR_WELD" present.

Mask values for bits 0 to 21 (LSB 0) are specified in the following table. Bits 22 to 127 are reserved.

Bit number (LSB 0)	Mask Value	Mask Name	Description
0	0x01	ERR_RCMB_TRIGGERED	Residual current detected via sen-sor.
1	0x02	ERR_VEHICLE_STATE_E	Vehicle signals error.
2	0x04	ERR_MODE3_DI-ODE_CHECK	Vehicle diode check failed - tamper detection.
3	0x08	ERR_MCB_TYPE2_TRIG-GERED	MCB of type 2 socket triggered.
4	0x10	ERR_MCB_SCHUKO_TRIG-GERED	MCB of domestic socket triggered.
5	0x20	ERR_RCD_TRIGGERED	RCD triggered.
6	0x40	ERR_CONTACTOR_WELD	Contactor welded.
7	0x80	ERR_BACKEND_DISCON-NECTED	Backend disconnected.
8	0x100	ERR_ACTUATOR_LOCK-ING_FAILED	Plug locking failed.
9	0x200	ERR_ACTUATOR_LOCK-ING_WITH-OUT_PLUG_FAILED	Locking without plug error.
10	0x400	ERR_ACTUATOR_STUCK	Actuator stuck cannot unlock.
11	0x800	ERR_ACTUATOR_DETEC-TION_FAILED	Actuator detection failed.
12	0x1000	ERR_FW_UPDATE_RUN-NING	FW Update in progress.
13	0x2000	ERR_TILT	The charge point is tilted.
14	0x4000	ERR_WRONG_CP_PR_WIR-ING	CP/PR wiring issue
15	0x8000	ERR_TYPE2_OVER-LOAD_THR_2	Car current overload, charging stopped.

16	0x10000	ERR_ACTUATOR_UN-LOCKED_WHILE_CHARG-ING	Actuator unlocked while charging.
17	0x20000	ERR_TILT_PRE-VENT_CHARGING_UN-TIL_REBOOT	The charge point was tilted and it is not allowed to charge until the charge point is rebooted.
18	0x40000	ERR_PIC24	PIC24 error.
19	0x80000	ERR_USB_STICK_HAN-DLING	USB stick handling in progress.
20	0x100000	ERR_INCOR-RECT_PHASE_INSTALLA-TION	Incorrect phase rotation direction detected.
21	0x200000	ERR_NO_POWER	No power on mains detected.