KeContact P20 / P30

UDP Programmers Guide

KEBA

Automation by innovation.

Comments to this manual

In this manual you will find warnings against possible dangerous situations. The used symbols apply to the following meanings:

Notice
Notes on use of equipment and useful practical tips are identified by “Notice". Notices do not contain any information that draws attention to potentially dangerous or harmful functions.

► Step of a sequence of operations.

Concerning this manual

This manual is intended for use by customers and personnel with the relevant technical knowledge in the applicable area and appropriate to the operations they are required to perform.

Target group:

- Customers/personnel with some programming knowledge

List of changes

Version: V2.00	Date: 20.10.2016
Page	**Description**
4	Changed chapter “Requirements”
5	Added chapter “Supported UDP commands”
7	Changed chapter “UDP command “failsafe”
8-9	Changed chapter “report”
11-17	Added new UDP commands: - report 1xx, currtime, setenergy - display, start, stop, unlock
-	Some minor changes and improvements (Internet links, frame style of images etc.)

Document: Revision 2.00 / Article no.: 92651

KEBA AG, Postfach 111, Gewerbepark Urfahr, A-4041 Linz , www.keba.com/emobility

Contents

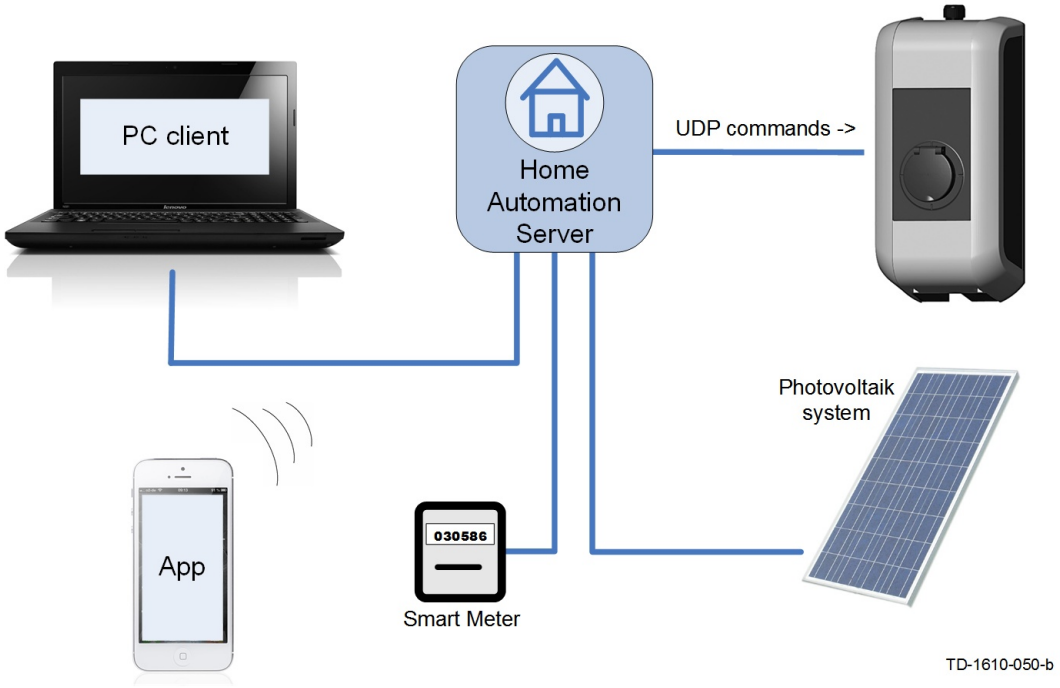
1 Introduction

The Programmers Guide provides the necessary information how to send UDP commands to your charging station. It gives you the possibility to get status information from the charging station or control it with an UDP Client (PC software or Smartphone App).

The **User Datagram Protocol (UDP)** is a simple network protocol that belongs to the transport layer of the Internet protocol family.

With suitable Home automation controllers it is also possible to realize e.g. a reduction of your energy consumption in conjunction with a photovoltaic system and your charging station.

Schematic overview (example)



1.1 Requirements

The following requirements have to be met to use the UDP functionality:

- KeContact P20 Charging station with network connection (LSA+ socket)
  - Product code: KC-P20-xxxxxx**2**x-xxx or KC-P20-xxxxxx**3**x-xxx
  - Firmware version: 2.5 or higher.
- KeContact P30 Charging station or BMW wallbox
  - Firmware version 3.05 of higher.
- Enabled UDP function with DIP-switch DWS1.3 = ON (for details about the DIP-switches please refer to the “Installation manual”).
- Router or Smart-Home Server with integrated Web Server (e.g. Loxone MiniServer).
- A PC or Smartphone for sending the UDP commands.
  This also requires a suitable UDP Client software or UDP App.

2 UDP commands

2.1 General information

When using UDP commands it is strongly recommended to protect your Smart-Home-Network on **port 7090** against unauthorized control or external attacks.

The charging station reacts on broadcasts and directly addressed commands. In a group of several charging stations, direct addressing should be used because otherwise all charging stations would react.

The commands are sent to the charging station as simple text commands (without end characters such as <CR> or <LF>).

The answers come as data packages with up to 512 bytes and are formatted in JSON standard; each data line within the structure is terminated with <LF> (0x0A).

Notes:
For the JSON format, open-source-parsers are available.

UDP does not support any error correction function. Please verify the effect of the sent commands yourself.

Port information:
Send port = UDP 7090
Receive port = UDP 7090

2.2 Supported UDP commands

UDP command	KeContact P20	KeContact P30	BMW wallbox
Broadcast Messages	✓	✓	✓
Command “`i`”	✓	✓	✓
Command “`failsafe`”	✓	✓	✓
Command “`report`”	✓	✓	✓
Command “`report 1xx`”	✗	✓	✓
Command “`ena`”	✓	✓	✓
Command “`curr`”	✓	✓	✓
Command “`currtime`”	✗	✓	✓
Command “`setenergy`”	✗	✓	✓
Command “`output`”	✓	✓	✗
Command “`start`”	✗	✓	✓
Command “`stop`”	✗	✓	✓
Command “`display`”	✗	✓	✗
Command “`unlock`”	✓	✓	✓

2.3 UDP Broadcast Messages

The broadcast messages are intended to avoid the permanent polling of the following described reports.

If there is a change of the status, the authorization, the enable input X1, the maximum possible current presets (temperature reduction), or an increase of the energy value a message is generated for the change.

For a detailed evaluation of the status, the corresponding report can be queried.

<table>
<tr><th>UDP broadcast:</th><th></th></tr>
<tr><td>Messages:</td><td>

```
{"State": 2}
{"Plug": 1}
{"Input": 0}
{"Enable sys": 1}
{"Max curr": 32000}
{"E pres": 999999}
```

</td></tr>
<tr><td>Description:</td><td>
"State" = Current state of the charging station
0 : starting
1 : not ready for charging; e.g. unplugged, X1 or "ena" not enabled,
RFID not enabled,.....
2 : ready for charging; waiting for EV charging request (S2)
3 : charging
4 : error
5 : authorization rejected

"Plug" = Current condition of the loading connection

<table>
<tr><td>0</td><td>unplugged</td><td></td><td></td></tr>
<tr><td>1</td><td>plugged on charging station</td><td></td><td></td></tr>
<tr><td>3</td><td>plugged on charging station</td><td>plug locked</td><td></td></tr>
<tr><td>5</td><td>plugged on charging station</td><td></td><td>plugged on EV</td></tr>
<tr><td>7</td><td>plugged on charging station</td><td>plug locked</td><td>plugged on EV</td></tr>
</table>
"Input" = State of the potential free Enable input X1.
When using the input, please pay attention to the information
in the installation manual.

"Enable sys" = Enable state for charging (contains Enable input, RFID, UDP,..).

"Max curr" = Current preset value via Control pilot in milliampere.

"E pres" = Power consumption of the current loading session in 0,1Wh;
Reset with new loading session (state = 2).

"E pres" is used as a pulse output with a valence 10Wh per message
or 100 messages / kWh -> full load 22kW
= one message every 1.6 sec
</td></tr>
</table>

2.4 UDP command “i”

UDP command:	`i`
Reply:	`"Firmware":"KEBA P20 v 2.5a3 (160613-061001):29309:143.0"`
Description:	Returns the product family and the firmware version of the charging station. Serves to a possible inventory of the network by means of a broadcast command with a subsequent report for detailed queries.

2.5 UDP command “failsafe”

UDP command:	`failsafe [Timeout] [curr] [save]`
Reply:	`"TCH-OK :done"`
Description:	[Timeout] 10...600 = Within this time, a command "ena" or "curr" must be received, in order not to activate the failsafe mode. 0 = Disables the failsafe mode. [curr] 0 = Disables the charging process when triggering the failsafe function, similar to command "ena 0". 6000...63000 = Reduces the maximum allowed charging current to this value (in milliampere), when triggering the failsafe function. [save] 0 = Activates the failsafe function only temporary until the next startup of the device. 1 = Saves the settings into the EEPROM in order to activate the failsafe function automatically, after restarting the device. By default, this feature is disabled and must be enabled once or after each restart. This function is used to limit a network charging station with a lost communication (network error, server failure, ...) to a defined state. This can be necessary in conjunction with a peak load limitation, in order not to overload the main fuses. An activated failsafe function is indicated by the charging station. In this case the LED segments do not longer illuminate the full width, but only indicates the current state with the middle two LED segments. A triggered failsafe function can be also recognized over UDP: in the report 2 the "`Enable sys`" value goes to 0, and the charge current setting "`Max curr`" goes to the default value of "`Curr FS`". After the failsafe function has been activated due to a timeout, you must set the values of current and the load enable again using the associated commands "curr" and "ena".

2.6 UDP command „report“

UDP command:	**report 1**

Reply P30:	{
	"ID": "1",
	"Product": "BMW-10-EC2405B2-E1R",
	"Serial": "17101357",
	"Firmware":"P30 v 3.05.1a5 (161020-130231)",
	"COM-module": 0,
	"Sec": 494
	}

Reply P20:	{
	"ID": "1",
	"Product": "KC-P20-ES240010-000 ",
	"Serial": "15017355",
	"Firmware":"KEBA P20 v 2.01m11 (140610-073512)"
	}

Description:	"ID" = ID of the retrieved report. "Product-ID" = Model name (variant) "Serial” = Serial number "Firmware" = Firmware version "COM-module” = Communication module is installed "Sec" = Current system clock since restart of the charging station.

UDP command:	**report 2**

Reply P30:	{
	"ID": "2",
	"State": 2,
	"Error1": 99,
	"Error2": 99,
	"Plug": 1,
	"Enable sys": 1,
	"Enable user": 1,
	"Max curr": 32000,
	"Max curr %": 1000,
	"Curr HW": 32000,
	"Curr user": 63000,
	"Curr FS": 63000,
	"Tmo FS": 0,
	"Curr timer": 0,
	"Tmo CT": 0,
	"Setenergy": 0,
	"Output": 0,
	"Input": 0,
	"Serial": "15017355",
	"Sec": 4294967296,
	}

Reply P20:

```
{
"ID": "2",
"State": 2,
"Error1": 99,
"Error2": 99,
"Plug": 1,
"Enable sys": 1,
"Enable user": 1,
"Max curr": 32000,
"Max curr %": 1000,
"Curr HW": 32000,
"Curr user": 63000,
"Curr FS": 63000,
"Tmo FS": 0,
"Output": 0,
"Input": 0,
"Serial": "15017355",
"Sec": 4294967296,
}
```

Description:

"ID" = ID of the retrieved report.

"State" = Current state of the charging station
- 0 : starting
- 1 : not ready for charging; e.g. unplugged, X1 or "ena" not enabled, RFID not enabled,.....
- 2 : ready for charging; waiting for EV charging request (S2)
- 3 : charging
- 4 : error
- 5 : authorization rejected

"Error 1" = Detail code for state 4; exceptions see FAQ on www.kecontact.com

"Error 2" = Detail code for state 4 exception #6 see FAQ on www.kecontact.com

"Plug" = Current condition of the loading connection

0	unplugged		
1	plugged on charging station		
3	plugged on charging station	plug locked	
5	plugged on charging station		plugged on EV
7	plugged on charging station	plug locked	plugged on EV

"Enable sys" = Enable state for charging (contains Enable input, RFID, UDP,..).

"Enable user" = Enable condition via UDP.

"Max curr" = Current preset value via Control pilot in milliampere.

"Max curr %" = Current preset value via Control pilot in 0,1% of the PWM value

"Curr HW" = Highest possible charging current of the charging connection. Contains device maximum, DIP-switch setting, cable coding and temperature reduction.

"Curr user" = Current preset value of the user via UDP; Default = 63000mA.

"Curr FS" = Current preset value for the Failsafe function.

"Tmo FS" = Communication timeout before triggering the Failsafe function.

"Curr timer" = Shows the current preset value of currtime.

"Tmo CT" = Shows the remaining time until the current value is accepted.

"Setenergy" = Shows the set energy limit.

"Output" = State of the output X2.

"Input" = State of the potential free Enable input X1. When using the input, please pay attention to the information in the installation manual.

"Serial" = Serial number

"Sec" = Current system clock since restart of the charging station.

UDP command:	`report 3`

Reply:	<pre>{
	"ID": "3",
	"U1": 230,
	"U2": 230,
	"U3": 230,
	"I1": 99999,
	"I2": 99999,
	"I3": 99999,
	"P": 99999999,
	"PF": 1000,
	"E pres": 999999,
	"E total": 9999999999,
	"Serial": "15017355",
	"Sec": 4294967296,
	}</pre>

Description:	"ID" = ID of the retrieved report. "U1"|"U2"|"U3"| = Current voltage in V. "I1"|"I2"|"I3"| = Current current value of the 3 phases in mA. "P" = Current power in mW (Real Power). "PF" = Power factor in 0,1% (cosphi) "E pres" = Power consumption of the current loading session in 0,1Wh; Reset with new loading session (state = 2). "total" = Total power consumption (persistent) without current loading session 0,1Wh; Is summed up after each completed charging session (state = 0). "Serial" = Serial number

2.7 UDP command “report 1xx” (historical log entries)

With the commands “report 101” up to “report 130” you can read the last 30 historical charging sessions. Report 100 shows the latest charging session. If no vehicle is connected this report delivers initially everything zero after a reboot of the charging station.
If a new session is started by plugging in the vehicle, a new Session ID is created in report 100 and all the starting values (start time, start energy value, rfid token code) are filled.
If the session ends, the end values (end time and end reason) will also be added, and the report 100 is shifted to report 101.

UDP command:	**`report 100` (example: after reboot, no vehicle connected)**

Reply:	`{`
	`"ID": "100",` // answer ID; is equal to calling report number
	`"Session ID": 0,` // running session counter; not resettable
	`"Curr HW  ": 0,` // maximum charging current of the cable and the charging station setting (equal to report 2)
	`"E Start  ": 0,` // total energy value at the beginning of the session
	`"E Pres  ": 0,` // delivered energy until now (equal to E pres in report 3)
	`"started[s]": 0,` // system time when the session was started (seconds from reboot; NTP implementation is still under progress)
	`"ended[s]  ": 0,` // system time when the session has ended
	`"reason ": 0,` // reason for stopping the session (1 = vehicle unplug; 10 = Rfid token)
	`"RFID tag": "0000000000000000",` // RFID Token ID if session started with rfid; hexadecimal; first character is the lowest nibble
	`"RFID class": "00000000000000000000",` // RFID classifier shows the defined color code if the used card is a BMW card (for example “010104” means the white card)
	`"Serial": "16914905",` // serial number of the charging station
	`"Sec": 111` // current time when the report was generated
	`}`

UDP command:	**`report 100` (example: authorized with RFID card, vehicle already charging)**

Reply:	`{`
	`"ID": "100",`
	`"Session ID": 34,`
	`"Curr HW  ": 20000,`
	`"E Start  ": 26731,`
	`"E Pres  ": 27242,`
	`"started[s]": 1065,`
	`"ended[s]  ": 0,`
	`"reason  ": 0,`
	`"RFID tag  ": "e3f76b8d00000000",`
	`"RFID class": "01010400000000000000",`
	`"Serial": "16914905",`
	`"Sec": 1179`
	`}`

UDP command:	**report 100** (example: vehicle unplugged)

Reply:	{
	"ID": "6",
	"Session ID": 34,
	"Curr HW ": 20000,
	"E Start ": 26731,
	"E Pres ": 29532,
	"started[s]": 1065,
	"ended[s] ": 1338,
	"reason ": 1,
	"RFID tag ": "e3f76b8d00000000",
	"RFID class": "01010400000000000000",
	"Serial": "16914905",
	"Sec": 1355
	}

UDP command:	**report 100** (example: authorized with RFID card, vehicle is already plugged in but not charging)

Reply:	{
	"ID": "100",
	"Session ID": 35,
	"Curr HW ": 20000,
	"E Start ": 29532,
	"E Pres ": 0,
	"started[s]": 1698,
	"ended[s] ": 0,
	"reason ": 0,
	"RFID tag ": "e3f76b8d00000000",
	"RFID class": "01010400000000000000",
	"Serial": "16914905",
	"Sec": 1704
	}

UDP command:	**report 122** (example of a historical log entry)

Reply:	{
	"ID": "122",
	"Session ID": 12,
	"Curr HW ": 20000,
	"E Start ": 18737,
	"E Pres ": 833,
	"started[s]": 351,
	"ended[s] ": 661,
	"reason ": 1,
	"RFID tag ": "e3f76b8d00000000",
	"RFID class": "01010400000000000000",
	"Serial": "16914905",
	"Sec": 916
	}

2.8 UDP command “ena”

UDP command:	`ena [n]`
Reply:	`"TCH-OK :done"`
Description:	[n] = Is the enable state (binary). 0 = Disabled; is indicated with a blue flashing LED. The same as used with the external enable input. 1 = Enabled The answer TCH-OK confirms only the receiving of the command and not the correctness of the value. A value unequal to 0 or 1 is also confirmed and is leading to deactivation. The correct reception can be verified using the proper report or by getting the corresponding broadcast message. ATTENTION: Some electric vehicles (EVs) do not yet meet the standard requirements and an "ena 0" can lead to an error in the charging station.

2.9 UDP command “curr”

UDP command:	`curr [n]`
Reply:	`"TCH-OK :done"`
Description:	[n] = Is the maximum allowed loading current in milliampere. The answer TCH-OK confirms only the receiving of the command and not the correctness of the value. Allowed are values between 6000mA and 63000mA. Invalid values are discarded and the default is set to 6000mA. The value is also depending on the DIP-switch settings and the used cable of the charging station. The correct reception can be verified using the proper report or by getting the corresponding broadcast message.

2.10 UDP command “currtime”

The commands “currtime”, “setenergy”, “start” and “stop” are designed specifically for the use with apps when there is no permanent network connection. The set values can be read with the command “report 2”, especially the remaining timer value.

UDP command:	**`currtime [current] [delay]`**
Reply:	`"TCH-OK :done"`
Description:	[current] = Allowed loading current in milliampere. (allowed values are between 6000 mA and 63000 mA). [delay] = Time delay in seconds (max. 860400 sec.). With the command “currtime” it is possible to set a time delayed “curr” and / or “ena” command. This command will be disabled automatically after unplugging your vehicle. You can additionally define the immediate start value using the known commands “ena” and “curr”. This means you can start immediately with 16 amps by sending “curr 16000” and reduce the current to 10A in 2 hours by sending “currtime 10000 7200”. It is also possible to send “ena 0” followed by “currtime 32000 7200” to start charging with 32 amps in 2 hours. **Examples:** `currtime 32000 3600` // Charging starts with a maximum current of 32A in 1 hour. `currtime 0 0` // Disables the delayed command. `currtime 0 3600` // ena 0 executed in 1 hour. `currtime 0 1` // ena 0 executed in 1 sec.; this can be used instead of an “ena 0” but will be resetted to “ena 1” and “curr63000” after unplugging the vehicle.

2.11 UDP command “setenergy”

The commands “currtime”, “setenergy”, “start” and “stop” are designed specifically for the use with apps when there is no permanent network connection. The set values can be read with the command “report 2”.

UDP command:	**`setenergy [Energy]`**
Reply:	`"TCH-OK :done"`
Description:	[Energy] = Charging energy limit in 0,1Wh. With the command “setenergy” it is possible to charge until the Value “E pres” shown with “report 3” reaches the energy limit. This command will be disabled automatically after unplugging your vehicle. With this command it is possible to define an energy amount combined with a “curr” and also a “currtime” command without the need for the App to stay connected. After reaching the energy limit the charging station reacts like a “ena 0”. After the pause it is possible to start the charging again without a limit sending an “ena 1”.It is also possible to delete an already set limit using “0” as energy limit. **Examples:** `setenergy 100000` // Charging with the specified current until 10kWh is reached. `setenergy 0` // Disables the energy limit.

2.12 UDP command “output”

UDP command:	`output [n]`
Reply:	`"TCH-OK :done"`
Description:	[n] = Is the output state of the relay terminal X2 under the connector panel cover of the charging station (binary value). 0 = open 1 = closed >=10 = Pulse output with the specified number of pulses (pulses / kWh) and is stored in the EEPROM; reasonably usable up to 150. The answer TCH-OK confirms only the receiving of the command and not the correctness of the value. Values between 2 and 9 are not answered (reserved functions). Invalid values (e.g. letters) are leading to a deactivation of the relay. The correct reception can be verified using the proper report.

Note
The controllability is only available if the default function of the output is not activated via the DIP-switch DSW1.2.

2.13 UDP command “start”

The commands “currtime”, “setenergy”, “start” and “stop” are designed specifically for the use with apps when there is no permanent network connection.

UDP command:	`start [RFID tag] [RFID classifier]`
Reply:	`"RFID tag": "e3f76b8d00000000",` `"RFID class": "0101040000000000000",`
Description:	[RFID tag] = 8 byte hex string. [RFID classifier]= 10 byte hex string. In this example “04” shows the color code of the card; (01=red, 02=green, 03=blue, 04=white) The Token (tag) and also the Classifier is a hexadecimal string like you can see it in the “report 100”. With this command you can remotely authorize the charging session using your App like swiping your RFID card at the reader. The Token and also the Classifier are optional. If you use an existing Token which was also teached on the charging station, you are also able to stop the charging session swiping the physical Token. You can extract the RFID Tokens and RFID Classifiers working on this charging station by parsing the historical logs “report 101” till “report 130”. By sending also the optional RFID classifier this information is only forwarded to the backend. You can also use your own pattern to remotely start a session. But be careful; you have to use the same pattern for remote stop and you have no possibility to stop it by physical card.

2.14 UDP command “stop”

The commands “currtime”, “setenergy”, “start” and “stop” are designed specifically for the use with apps when there is no permanent network connection.

UDP command:	`stop [RFID Token]`
Reply:	`"TCH-OK :done"`
Description:	[RFID Token] = The token is a hexadecimal string like you can see it in “report 100”. With this command you remotely stop a charging session using your App like swiping your RFID card at the reader. For doing this, you can use the authorization information presented with “report 100”. This command shows the same audio visual feedback on the charging station like you are swiping the RFID card at the reader.

2.15 UDP command “display” (for P30 with display only)

UDP command:	`display [a] [min] [max] [tk] [text]`
Reply:	`"TCH-OK :done"`
Description:	[a] = Activates the usage of none default values for the time [text] is displayed. 0 = default (min-time = 2 sec, max-time = 10 sec, fixed token) 1 = min and max time can be defined [min] = Defines the duration in seconds how long the text will displayed before another display command will be processed (internal MID metering relevant information may overrule this). [max] = Defines the duration in seconds how long the text will displayed if no additional display command follows. [tk] = Token – for internal use only. [text] = Text shown on the display. Maximum 23 ASCII characters can be used. 0 .. 23 characters ~ == Σ $ == blank , == comma **Note:** If you use the text “kWh”, it will be replaced with “???” (due to MID metering certification). **Examples:** `display 0 0 0 0 12346,6`

2.16 UDP command “unlock”

UDP command:	`unlock`
Reply:	`"TCH-OK :done"`
Description:	For this command you have to stop the charging process first. For this, please use the command „ena 0“. Afterwards you can unlock the socket.

3 Further informations

Loxone MiniServer (example)

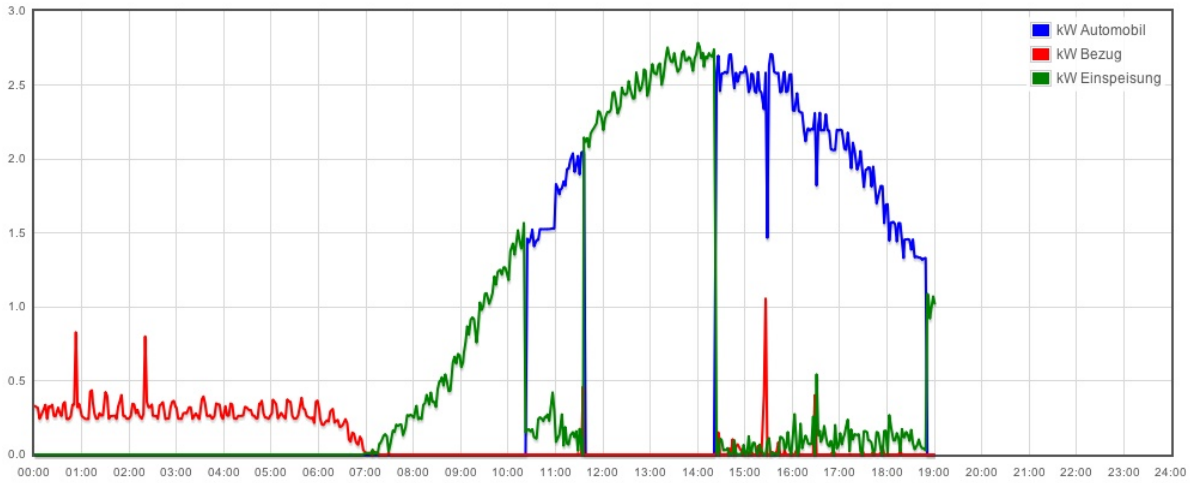
Loxone MiniServer

The UDP features are tested with Loxone MiniServer for house automations. For details please see www.loxone.com .

For the installation please follow the detailed instructions under the documentation section “Loxone config -> Communication with UDP”.

Software solution for energy optimization

This is an existing Open-Source implementation for optimizing the excess energy of a photovoltaic system combined with KeContact UDP-interface and other KNX-based systems for Linux x86 and ARM (Raspberry Pi).
http://www.eb-systeme.de



Power consumption

Legend	
blue	EV consumption
red	Complete consumption (household)
green	Total excess energy (household)

(Page left blank)

www.keba.com/emobility