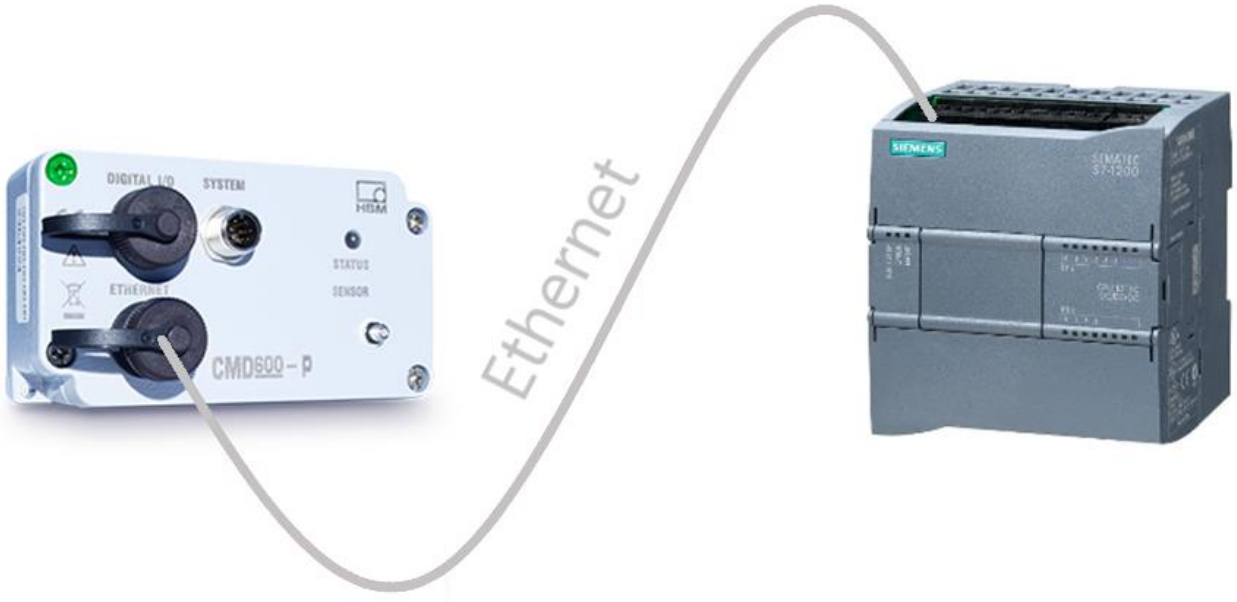
TECH NOTE – Integrate CMD Charge Amplifier in Profinet-networks (with TIA portal)

Version: 2019-07-24
Author: Michael Guckes
Status: HBM: Public

Brief description

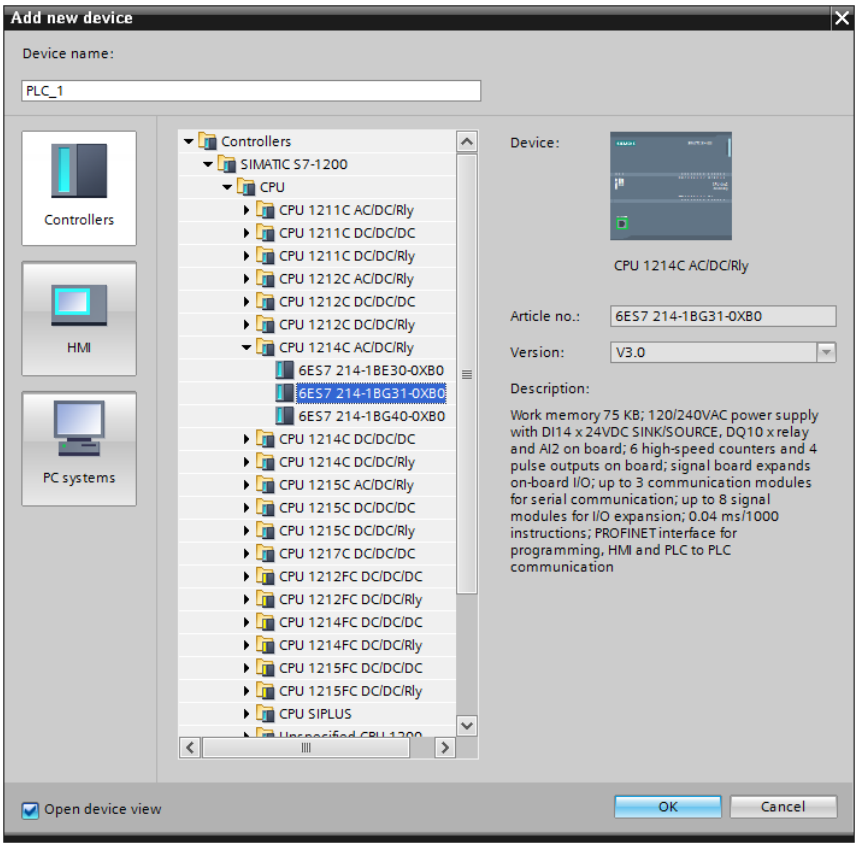
This is an instruction for using a CMD charge amplifier with a Siemens 1200 controller. In this example it is explained how to stream data via UDP from the charge amplifier to the controller via UDP and send a command via TCP. This non-real-time communication can be performed simultaneously to an existing real-time ProfiNet connection over the same line. In this example the communication is done with program blocks, which are enclosed with the technote in the .scl file.



Project setup in TIA Portal and block importing

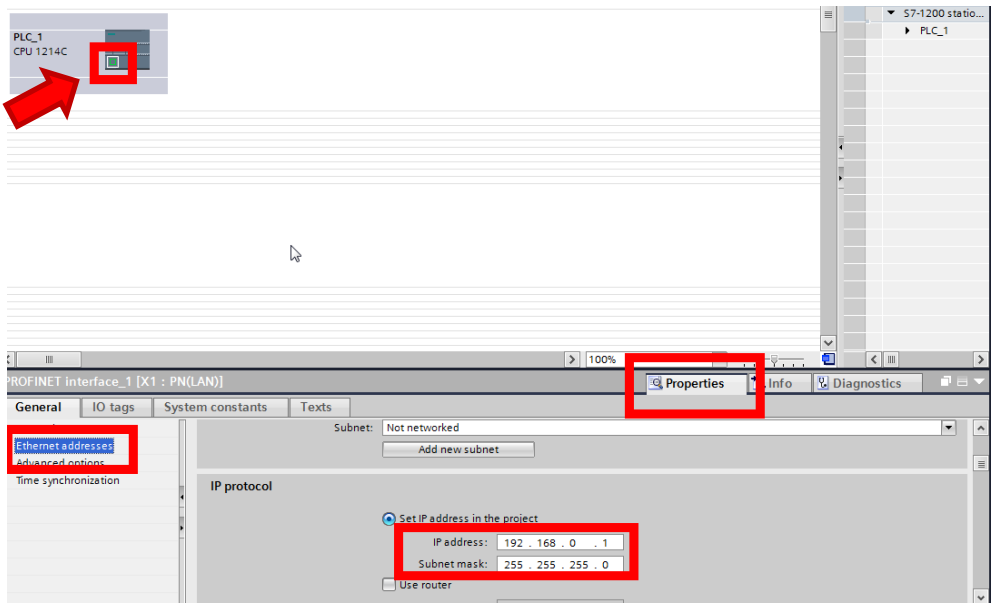
Start TIA Portal and create a project.

- Assign a name to your project
- Select ‘Devices & networks’ and add a new device
- Add the PLC used (here: Simatic 1200 Station) → label on the right side of the PLC; firmware version on the sticker



Network configuration

- Switch to ‘Network View’ – the PLC should be visible here

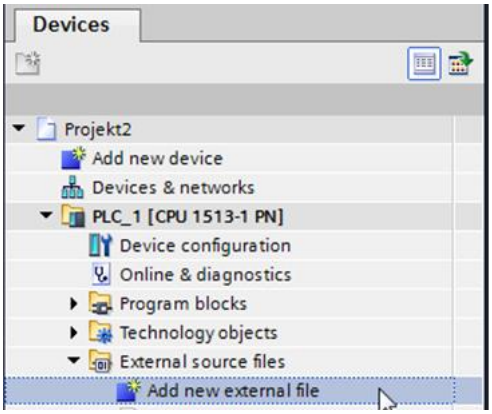


- Select the device by clicking on the green square
- Select properties → Ethernet addresses
- Adapt the IP address to the one of your device

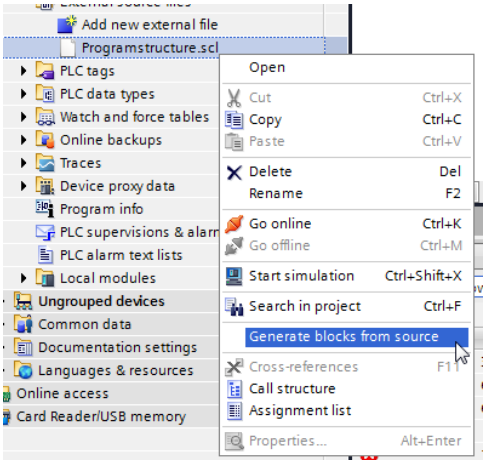
Importing the blocks

After the device is set up the function blocks need to be imported. The function blocks contain the functionality of receiving data via UDP, send data via TCP and the associated setting files.

- Double click ‘Add new external file’
  (External source files>Add external file).
- Select the CMDProgram.scl file



- Choose the file ‘programstructure.scl’.
- Right click on the file and click on ‘Generate blocks from source’.

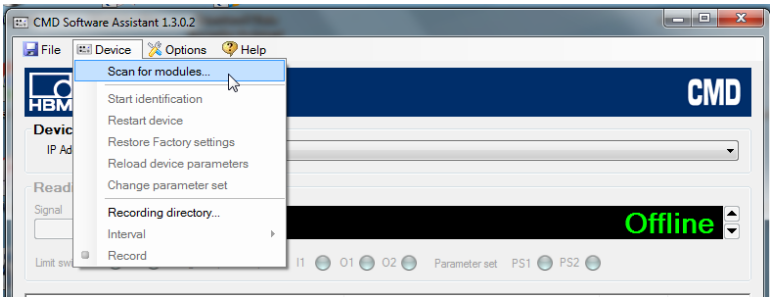


After executing the folder program blocks contains all the necessary blocks. The program can now be loaded into the PLC.

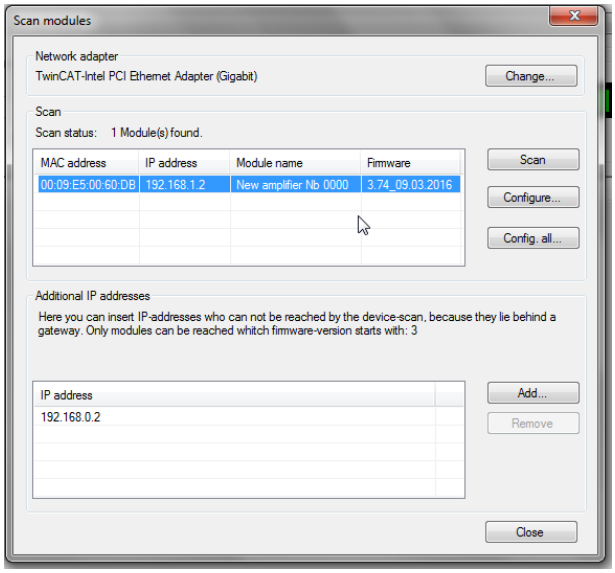
Setting up the CMD

- Run the CMD Software Assistant
- The CMD charge amplifier can be found using the scan functionality of the assistant

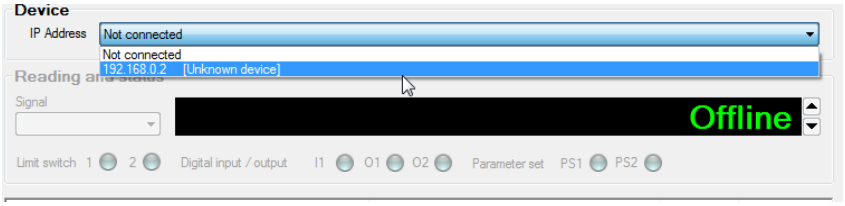
For the CMD Setup please download the CMD Assistant from https://www.hbm.com/en/2659/paceline-cmd-digital-charge-amplifier/



- The amplifier should now appear in the List of available devices.



- Click ‘Close’
- Connect to the charge amplifier by selecting it in the drop-down menu

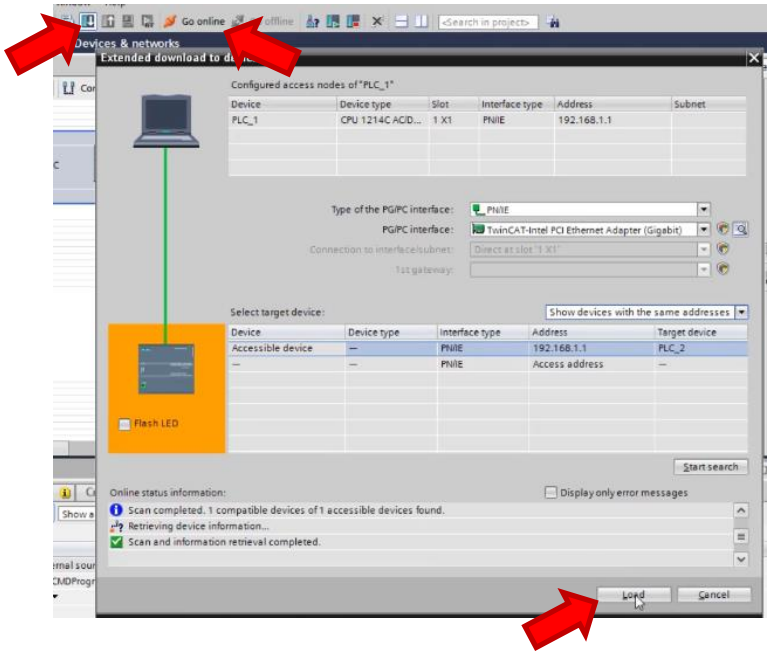


As soon as you are connected, make sure that the PLC and the CMD charge amplifier are in the same network and use the same IP address space. If this is not the case the IP address space of the CMD should be adapted to match the one of the PLC. After the configuration switch to ‘SignalOutput’ → ‘Digital’ and fill in the corresponding IP address of the PLC. The default port of the program is 80. To activate streaming use the drop-down menu at “Activate” and set it to Enabled.

Now open the web interface of ClipX and activate the ProfiNet interface and enter the network settings.

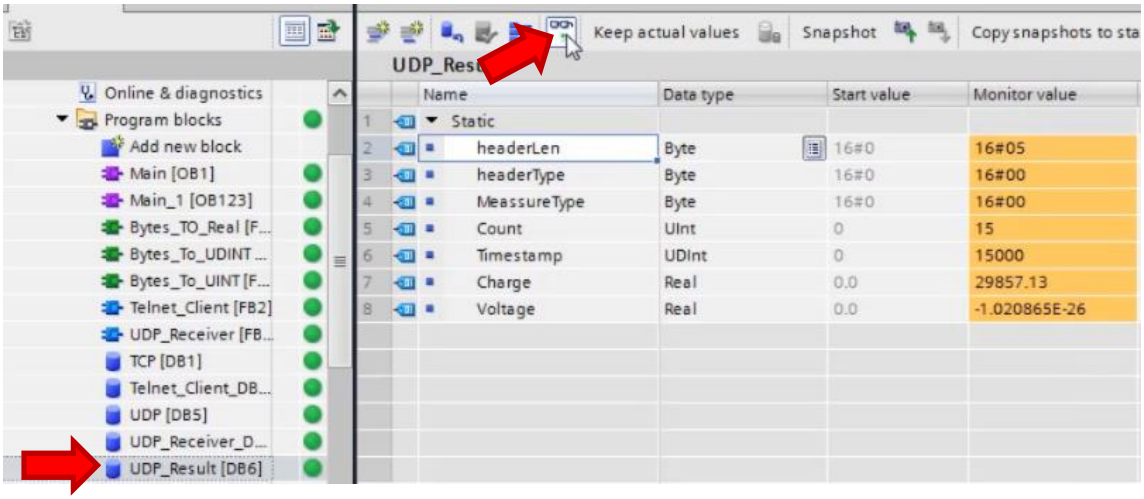
Program structure

- Click the download icon, load the program and click ‘Go online’



UDP Streaming:

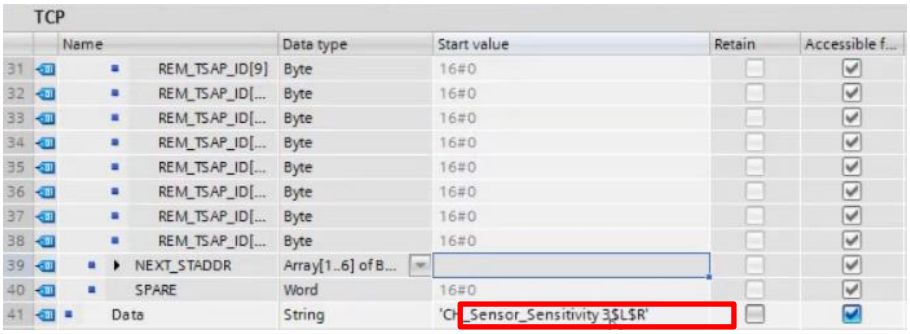
- Switch to the data base ‘UDP Result’ and activate the live monitoring



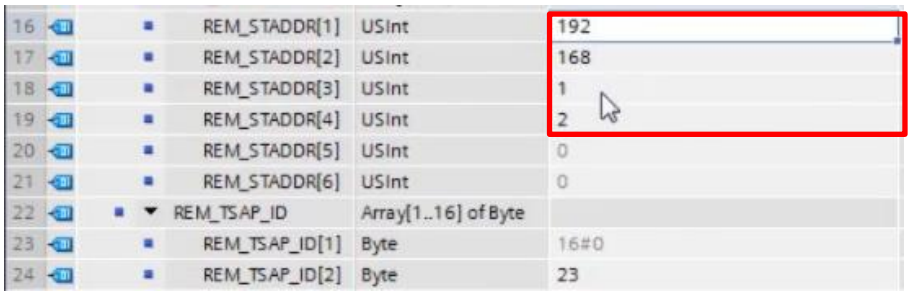
Now the live values from the CMD can be monitored and processed.

TCP Command:

- Switch to the data base ‘TCP’ and adapt the last entry to the command that should be written (in this example the sensitivity of the CMD is adapted, see chapter program structure)



- Enter the IP address of the CMD



- Activate the ‘Telnet_Client’ method and load, like it was done in the previous chapter, to the PLC and start it

The program can be stopped after a few seconds. The command should be written.

Program structure

The program consists of the following components:

- **Main_1:**
  This routine is called cyclically. You can select whether you want to send commands to the CMD by calling Telnet_Client or if you just want to receive the streamed data by calling UDP_Receiver.

- **TCP:**
  This data block stores the TCP settings.

TCP

	Name	Datentyp	Startwert	Remanenz	Erreichbar a..	Schrei...	Sichtbar i...	Einstellwert	Kommentar
13	LOCAL_TSAP_ID	Array[1..16] of Byte			☑	☑	☑		TSAP id/local port number
14	REM_SUBNET_ID	Array[1..6] of USInt			☑	☑	☑		remote subnet id
15	REM_STADDR	Array[1..6] of USInt			☑	☑	☑		remote IP address
16	REM_STADDR[1]	USInt	192		☑	☑	☑		remote IP address
17	REM_STADDR[2]	USInt	168		☑	☑	☑		remote IP address
18	REM_STADDR[3]	USInt	1		☑	☑	☑		remote IP address
19	REM_STADDR[4]	USInt	100		☑	☑	☑		remote IP address
20	REM_STADDR[5]	USInt	0		☑	☑	☑		remote IP address
21	REM_STADDR[6]	USInt	0		☑	☑	☑		remote IP address
22	REM_TSAP_ID	Array[1..16] of Byte			☑	☑	☑		TSAP id/remote port number
23	REM_TSAP_ID[1]	Byte	16#0		☑	☑	☑		TSAP id/remote port number
24	REM_TSAP_ID[2]	Byte	200		☑	☑	☑		TSAP id/remote port number
25	REM_TSAP_ID[3]	Byte	16#0		☑	☑	☑		TSAP id/remote port number
26	REM_TSAP_ID[4]	Byte	16#0		☑	☑	☑		TSAP id/remote port number

The remote address must be updated to match the one of the CMD charge amplifier and the port is fixed at 23 because Telnet protocol is used for communication. The available commands can be found in the Interface-description of the CMD.

Important: Every command must be ended with a 'Carriage Return/Linefeed', otherwise the command is only buffered and not executed. Tia Portal uses the literals 'LR' for achieving this.
The local port can be anything between 100-65000:

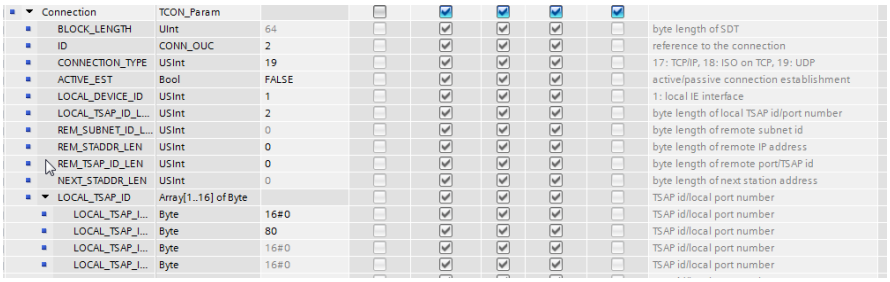
	NEXT_STADDR_LEN	USInt	0		☑	☑	☑		Byte length of next station address
	LOCAL_TSAP_ID	Array[1..16] of Byte			☑	☑	☑		TSAP id/local port number
	LOCAL_TSAP_I...	Byte	0		☑	☑	☑		TSAP id/local port number
	LOCAL_TSAP_I...	Byte	200		☑	☑	☑		TSAP id/local port number
	LOCAL_TSAP_I...	Byte	16#0		☑	☑	☑		TSAP id/local port number

In our example Port 200 is used.

- **Telnet_Client:**
  This function block is used to communicate with the charge amplifier. Therefore, it uses the command T_Send_c. Please find further details in the Tia documentation.

- **Udp_Receiver:**
  This function block opens a socket on the port defined in the UDP data block and processes received packets. The function block receives the UDP streamed data from the charge amplifier, processes them and stores them in the structure UDP_result. It is also able to send UDP packets to other participants in state 8, which is not necessary in this example.

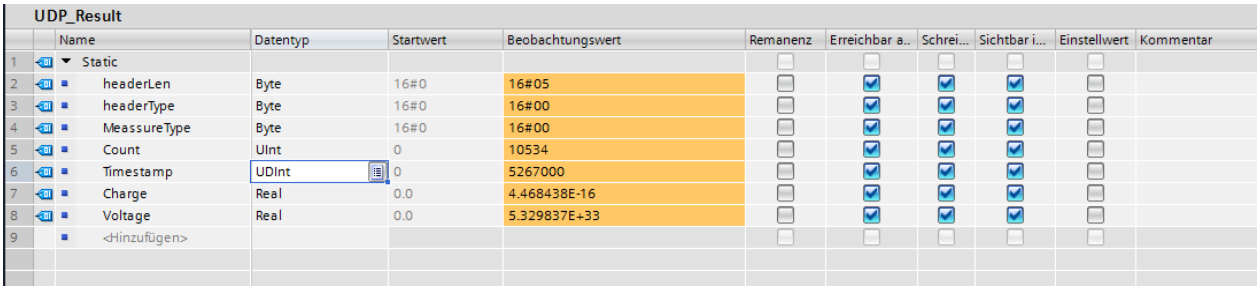
- **UDP:**
  This data block saves UDP settings.



Connection	TCON_Param								
BLOCK_LENGTH	UInt	64							byte length of SDT
ID	CONN_OUC	2							reference to the connection
CONNECTION_TYPE	USInt	19							17: TCP/IP, 18: ISO on TCP, 19: UDP
ACTIVE_EST	Bool	FALSE							active/passive connection establishment
LOCAL_DEVICE_ID	USInt	1							1: local IE interface
LOCAL_TSAP_ID_L...	USInt	2							byte length of local TSAP id/port number
REM_SUBNET_ID_L...	USInt	0							byte length of remote subnet id
REM_STADDR_LEN	USInt	0							byte length of remote IP address
REM_TSAP_ID_LEN	USInt	0							byte length of remote port/TSAP id
NEXT_STADDR_LEN	USInt	0							byte length of next station address
LOCAL_TSAP_ID	Array[1..16] of Byte								TSAP id/local port number
LOCAL_TSAP_I...	Byte	16#0							TSAP id/local port number
LOCAL_TSAP_I...	Byte	80							TSAP id/local port number
LOCAL_TSAP_I...	Byte	16#0							TSAP id/local port number
LOCAL_TSAP_I...	Byte	16#0							TSAP id/local port number

  The local port must match the one selected in the CMD assistant for UDP-Streaming.

- **UDP_Result:**



UDP_Result

	Name	Datentyp	Startwert	Beobachtungswert	Remanenz	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar
1	Static									
2	headerLen	Byte	16#0	16#05						
3	headerType	Byte	16#0	16#00						
4	MeassureType	Byte	16#0	16#00						
5	Count	UInt	0	10534						
6	Timestamp	UDInt	0	5267000						
7	Charge	Real	0.0	4.468438E-16						
8	Voltage	Real	0.0	5.329837E+33						
9	<Hinzufügen>									

UDP_Result stores the last received data packet. The header size is constant 5. The header type and the measurement type are always 0. Count contains a consecutive number which is increased by 1 with each UDP packet. The timestamp indicates a time delta since the start of the streaming and charge and voltage contain the measured values as float values

Disclaimer

These examples are for illustrative purposes only. They cannot be used as the basis for any warranty or liability claims.