

## UDP Packet Description

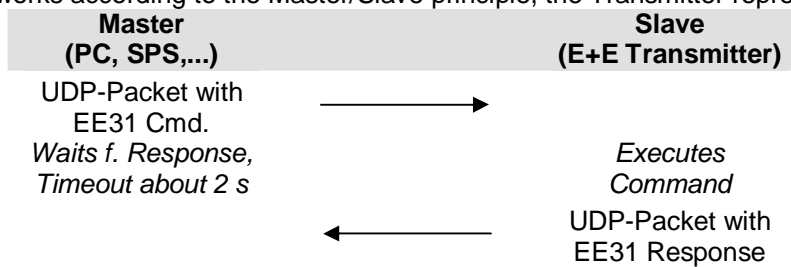
for Ethernet communication with E+E Transmitters

**UDP port number: 5234**

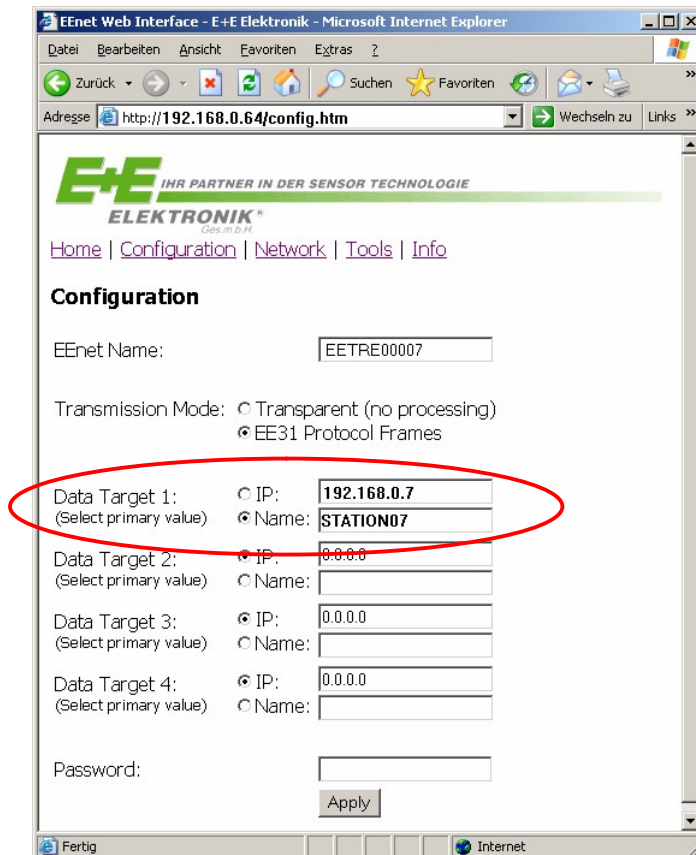
Communication over Ethernet is implemented using UDP packets, which carry a command or a command response (encoded in EE31 protocol format) as payload.

**Note:** The EE31 protocol format is described in detail in another document.

Communication works according to the Master/Slave principle, the Transmitter represents the Slave:



**Note:** In order to receive responses from the Transmitter, you have to configure the IP-Address (or DNS Name) of the Master in one of the four Data Target sections of the Transmitter. This can be done using the Web-Interface on page „Configuration“, input area „Data Target n“:



## Structure of UDP Packets

Master (PC, SPS,...) → Slave (E+E Transmitter):

Length Bytes	Data Type	Meaning
4	char [4]	Constant value „eEnT“ (ASCII representation, case sensitive!)
1	unsig. char	Constant value, decimal <b>99</b>
1	unsig. char	Constant value, decimal 0
2	unsig. short	Your software version, Major part, e.g. decimal 1
2	unsig. short	Your software version, Minor part, e.g. decimal 0
2	unsig. short	Your software version, Patch part, e.g. decimal 0
2	unsig. short	Your software version, Build part, e.g. decimal 1
1	unsig. char	Constant value, decimal 0
1	unsig. char	Constant value, decimal 0
1	unsig. char	Constant value, decimal <b>50</b>
1	unsig. char	Constant value, decimal 0
1	unsig. char	Constant value, decimal 0
1	unsig. char	Constant value, decimal 0
2	unsig. short	Total count of payload data bytes (EE31 protocol format)
4	char [4]	Constant value „EeNt“ (ASCII representation, case sensitive!)
???	unsig. char	Payload data bytes in EE31 protocol format

Slave (E+E Transmitter) → Master (PC, SPS,...):

Length Bytes	Data Types	Meaning
4	char [4]	Constant value, <b>must be „eEnT“</b> (ASCII representation)
1	unsig. char	Constant value, <b>must be between 1 and 98</b> (decimal)
1	unsig. char	Constant value, decimal 0
2	unsig. short	Ethernet module Firmware version, Major part, e.g. decimal 1
2	unsig. short	Ethernet module Firmware version, Minor part, e.g. decimal 0
2	unsig. short	Ethernet module Firmware version, Patch part, e.g. decimal 0
2	unsig. short	Ethernet module Firmware version, Build part, e.g. decimal 1
1	unsig. char	reserved, any value
1	unsig. char	reserved, any value
1	unsig. char	Constant value, <b>must be 30</b> (decimal)
1	unsig. char	reserved, any value
1	unsig. char	reserved, any value
1	unsig. char	reserved, any value
2	unsig. short	Total count of payload data bytes (EE31 protocol format)
4	char [4]	Constant value, <b>must be „EeNt“</b> (ASCII representation)
???	unsig. char	Payload data bytes in EE31 protocol format

## Description of Data Types

Data Type	Also known as	Value range
char	signed char	-128 to 127 (decimal, used for ASCII representation)
unsigned char	BYTE	0 to 255 (decimal)
unsigned short	WORD	0 to 65535 (decimal)

Data types, which consist of two or more bytes (int, float), are arranged in **Little Endian Byte Order** (Intel x86, least significant byte first, then upwards until most significant byte, e.g. 16-bit Integer value decimal: **16 0** hex: 0x**0010 0** sending order: 0x**10 00**).

	Datum	Unterschrift
erstellt:	15.03.2006	Helmut Girtzer
geprüft:		
freigegeben:		