EN: This Datasheet/Document is presented by the manufacturer.

Please visit our website for pricing and availability at www.hestore.hu.

Technical Data Sheet

Ethernet to Serial Interface Converter

COTER-E2I
COTER-E4I



Document version: COTER-E-man-en-v5.doc
Data of last revision: 2009-04-24
http://www.netronix.pl/

Contents

1. Features 5

2. Equipment 6

3. Element location in housing 7

4. Wiring diagram 8

5. Modes of operation 9

6. Start-up 10

6.1. Description of connection establishing using „Framer" program 10

6.2. Starting the NVT protocol 11

6.3. Connection with conversion server 12

6.3.1. RAW_MODE 12

6.3.2. NVT_MODE 13

7. Data transmission format using configuration server 15

8. Converter configuration 15

8.1. Network configuration 15

8.1.1. Physical address changing 15

8.1.2. Reading the physical address MAC 16

8.1.3. Setting the network parameters 16

8.1.4. Reading out the network parameters 17

8.1.5. Changing the network name 18

8.1.6. Reading out the network name 19

8.2. Controlling the access 19

8.2.1. Changing the settings of ACL list 20

8.2.2. Reading out the ACL list 20

8.2.3. Login 21

8.2.4. Log-out 22

8.2.5. Changing the password 22

8.3. Configuring the conversion server 23

8.3.1. Changing the settings of conversion server 23

8.3.2. Reading out the settings of conversion server 24

8.4. Controlling the I/O ports 25

8.4.1. Saving the settings of I/O ports 25

8.4.2. Reading out the I/O settings 26

8.4.3. Changing the status of an output port 26

8.4.4. Reading out the input status 27

8.5. General purpose commands 27
8.5.1. Inquiry on status 27
8.5.2. Reset command 28
8.5.3. Software version 28
8.5.4. Bootloader activation command 29

9. NVT protocol description 30

9.1. Short description of protocol 30
9.1.1. Commands implemented from RFC-854 standard 30
9.1.2. Commands interpreted from RFC-2217 protocol standard 31
9.1.3. Controlling the lines of I/O ports 32
9.2. Example of communication between client and server 33
9.2.1. Negotiating the service of „Telnet Com Port Control Option” 33
9.2.2. Querying on device signature 33
9.2.3. Changing the baud rate or serial interface 33
9.2.4. Changing the size of serial interface data frame 34
9.2.5. Changing the parity bit of serial interface 34
9.2.6. Changing the flow control, changing the RTS and DTR line status 35
9.2.7. Changing the status of I/O port 35

10. Description of signal on connectors 37

11. Electrical specifications 40

Index of illustrations

Picture 1: Housing elements 7
Picture 2: Example of wiring diagram 8
Picture 3: Changing the converter mode for NVT_MODE 12
Picture 4: Connection established in RAW_MODE 13
Picture 5: Connection in NVT_MODE using virtual port 14
Picture 6: Ethernet connector 37
Picture 7: RS-458, supply connector 38
Picture 8: RS-232, I/O, supply connector 39

Index of tables

Table 1: Housing elements 7
Table 2: Supported commands from RFC-854 standard 30
Table 3: Using "Com Port Option" from RFC-2217 protocol 31
Table 4: Extension of RFC-2217 protocol for controlling I/O ports 32
Table 5: Parameters of CAS_SET_GPIO option 32
Table 6: Parameters of CAS_SET_REG_GPIO option 33
Table 7: Signals on Ethernet connector 37
Table 8: LED’s in Ethernet connector 37
Table 9: Signals in RS-485, supply connector 38
Table 10: Signals in RS-232, I/O, supply connector 38
Table 11: Electrical parameters 40

1. Features

COTER-E family devises convert signals transmitted via Ethernet to serial interface signals.

- ✓ COTER-E4I converts Ethernet to RS-485.
- ✓ COTER-E2I converts Ethernet to RS-232 with possibility of controlling two I/O ports.

RS-232 version of the device is equipped with two general purpose I/O ports, by means of which it is possible to control remotely or check their current status, depending on their mode of configuration.

COTER-E converter can operate in three mode of communication:

- ✓ so-called OFF_MODE, where conversion server is switched off completely,
- ✓ so-called RAW_MODE, where data are sent directly with constant transmission parameters, fixed by means of configuration server,
- ✓ so-called NVT_MODE, where communication port is shared between data transfer and communication command transfer using protocol described in RFC-2217.

The device has Ethernet interface with configurable network parameters or with automatic download of DHCP server settings. By means of TCP/IP network connection, a user (host application) is able to modify settings or to read current device status out. To limit an access to devices connected to serial interface bus against intruders, COTER-E has ACL address list, from which it is possible to establish a communication. There is possibility to define login password allowing modifying settings too. The converter is equipped with feature searching for the device, which is present in local network. This feature is mainly useful during operation with active DHCP client. COTER-E family devices have implemented firmware BOOTLOADER, which can be preloaded by any TFTP client.

2. Equipment

- **1 Ethernet port**
  - ✓ Sending and receiving data to/from RS-232/485 interface.
  - ✓ Checking of I/O (1) status.
  - ✓ Configuring operation parameters of device.
  - ✓ Changing the firmware.

- **1 RS-232/RS-485 port**
  - ✓ Serial interface, transmission parameters are configurable by means of configuration server or using TELNET (RFC-2217) command option or using data conversion server.

- **2 IO (1) ports**
  - ✓ Controlling and checking of state remotely by means of configuration port or using port used for data transmission (extension RFC-2217).
  - ✓ Configuring the operation direction of I/O ports using configuration server only.

- **"Power supply" LED**
  - ✓ Diode signaling presence of converter supply voltage.

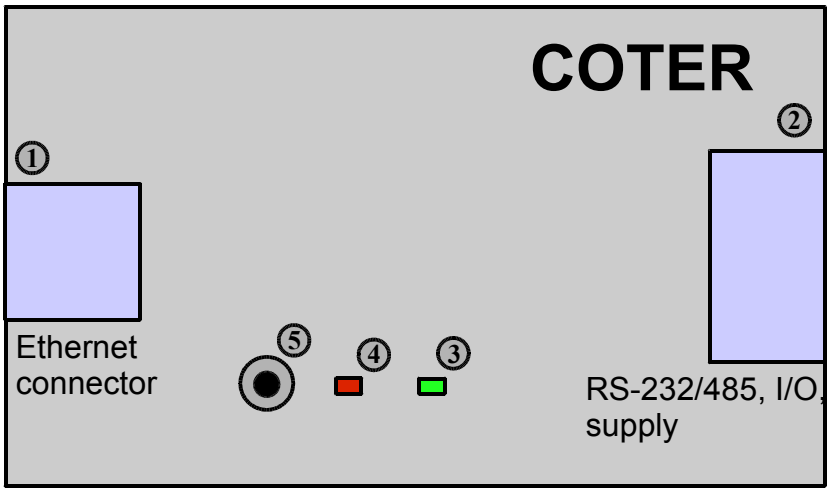
- **"Status" LED**
  - ✓ Diode signaling transmission via serial interface port.

- **"User" key**
  - ✓ Key for restoring the defaults.

Notes:

(1) Depending on converter version (see chapter Features).

3. Element location in housing



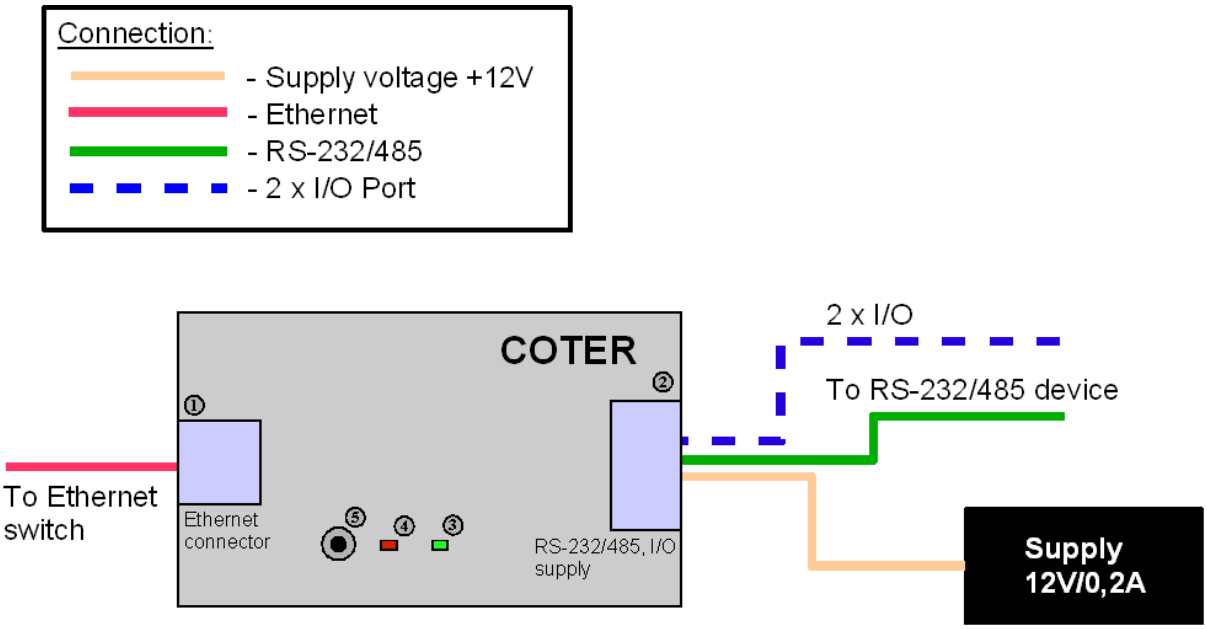
Picture 1: Housing elements

Table 1: Housing elements

No.	Name	Description
1	Ethernet jack	RJ45
2	RS-485, Power supply (1) jack	RJ-12 (supply voltage +8 to +16 V DC 300 mA max.).
2	RS-232, 2 x I/O, Power supply (1) jack	DB-9 (supply voltage +8 to +16 V DC 300 mA max).
3	"Power Supply" LED	Green.
4	"Status" LED	Red.
5	User key	-

Notes:
(1) Depending on converter version (see chapter Features).

4. Wiring diagram



Picture 2: Example of wiring diagram

5. Modes of operation

COTER-E converter has server implemented, which operates on one of TCP/IP ports (by defaults it is port 5000), to transmit the data. A device or PC must be connected to converter on communication port mentioned before. After establishing the communication, data which is sent to converter Ethernet port is transmitted to serial interface, and data which is received is sent immediately to host, from which the connection has been established.

There are three modes of operation, in which can operate a server, and which are used for data conversion:

1. OFF_MODE – data transmission via RS-232/485 <->Ethernet is switched off. In case of current connection with server which is present, entering the converter in this mode breaks the connection immediately off.
2. RAW_MODE – data transmission via RS-232/485<-> Ethernet directly, it means, that data received by the converter in both direction is sent directly to communication port, which the device communicates with. In this mode, changing the parameters of serial interface is possible to achieve by means of configuration server.
3. NVT_MODE – data transmission via RS-232/485<->Ethernet is carrying on same port as serial interface parameter configuration. Configuration data is sent using "Telnet Com Port Control Option" control protocol described in RFC-2217 document. In this mode, hardware control of data flow can be achieved by means of RTS/CTS (output control) and DTR/DSR (input control) signals. In case of immense data exchanging rate, using the communication with hardware flow control is advised, to avoid data corruption caused by serial interface line read-out buffer overflow.

6. Start-up

To activate the converter, connect to it power supply, serial interface and input proper signals from serial interface with respect to Table 9 or Table 10 depending on version.

The converter is configured by defaults for using the network settings download from DHCP sever. In case no information available on current network settings, it is possible to achieve it by sending advertising message with letter “D” to the port 30303. The converter will retrieve information with network name (by defaults: e.g. COTER-E2-V1 depending on device version and current configuration), MAC physical address (by defaults 00-04-A3-00-00-00) and with received IP address (by defaults 10.0.0.205) on message sender address, to 30303 port. If DHCP server is inactive, or from some reasons, the IP address is not assigned to the converter, as an IP address a constant value will be assigned which is saved in non-volatile memory. At the beginning it is value of 10.0.0.205, which can be changed using configuration server and command "SetInterfaceConfig" (see chapter network configuration). With sever DHCP disabled, the parameters which have been set using configuration server would be treated as valid and by means of these parameters converter would communicate in network.

To connect to device, establish TCP connection to port 50. If the device is not protected with password and ACL list (by defaults these protection is not active), everyone who will connect to server and uses NETRONIX protocol is able to modify the settings. If ACL list is not empty, connection can be established by means of selected addresses saved in converter memory only. If device password is active, modification of settings can be established, after previous login (sending proper frame with LoginUser command and valid password). Otherwise, it won’t be possible to modify settings, but verify them only. If remote host connected does not send any commands during 60 seconds, user will be logged out automatically.

Description of transmission protocol format is available on website "Netronix" company with short form description of FRAMER testing program.

6.1. Description of connection establishing using „Framer” program

To establish connection, perform following steps of the procedure:

- Check address IP of a device actually assigned, sending an advertising message with letter “D” to port 30303. It is possible to use Netronix Ethernet Discoverer" software available on manufacturer web site.
- Start "Framer" program. In dialog window Communication->Set switch of communication type to TCP/IP. Set connection type to "Client" and write in suitable bookmark correct IP address (e.g.: 10.0.0.205) and port 50. Press OK button. The program will try to establish a connection with converter, located on selected address. Connection status is given on a bar visible on right side. When connection will be established, announcement "Client: Connected" appears.
- Next load command file "frm" via File->Open menu and select suitable track to "COTER-E.frm" file. In this file there is included information on commands which are available for given converter. After loading the "frm" file, the commands which are available for converter can be selected in “Command” pulled-down list located on top of main window of the program.
- Select the command "C_FirmwareVersion" with code 0xFE from “Command” list of available commands. Send the command by pressing Enter key on keyboard, when indicator is in command edit window, or copying command into list using "Copy to list", indicate this command by clicking left key of a mouse and holding at the same

time key "Send". Converter responds for command sent returning as parameter number of software version installed. In lower part of “Monitor” main window there is transmission history in which are showed frames sent and received, depending on selected options.

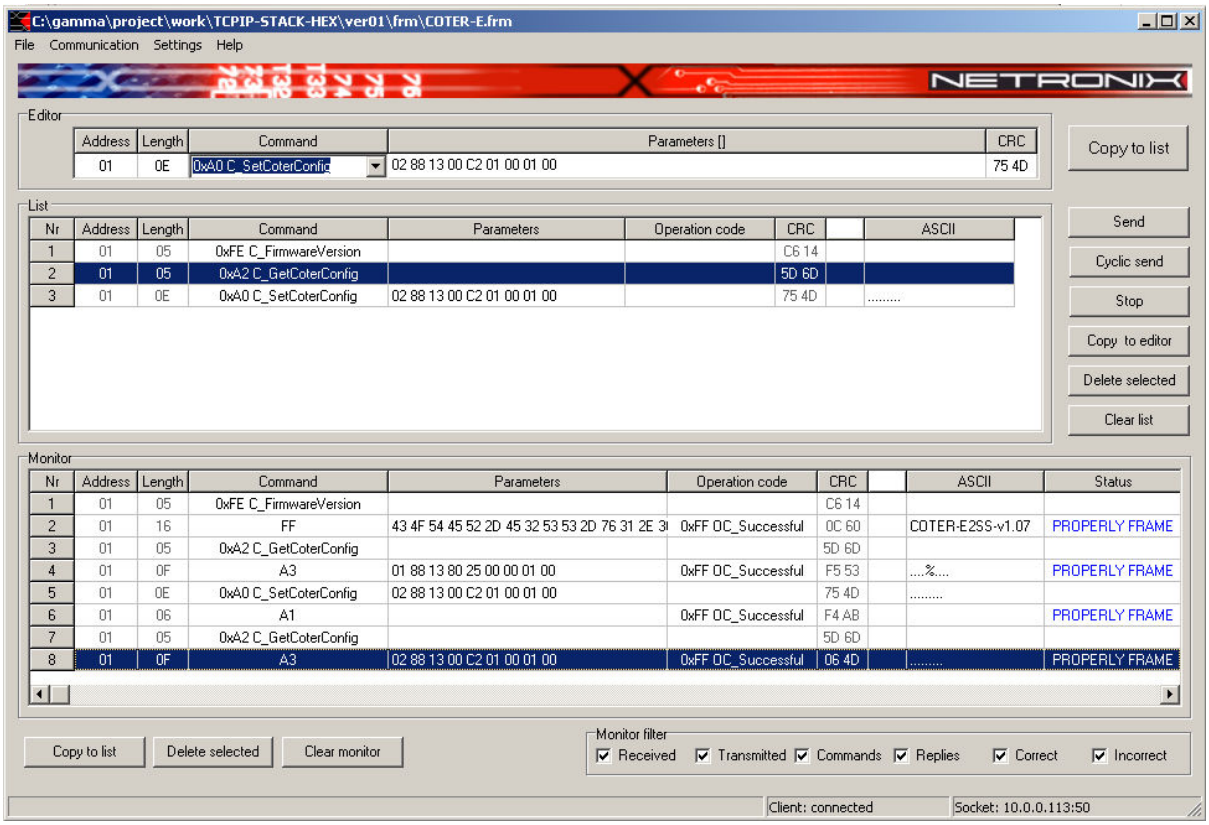
6.2. Starting the NVT protocol

By defaults, the converter operates in mode RAW_MODE which is active. To change the converter mode for NVT_MODE, connect to the converter in the way described in previous section using FRAMER program.

Perform following operations:

- Send "GetCoterConfig" command, to verify current converter settings. In default configuration, retrieved parameters should be of: 01 88 13 80 25 00 00 01 00 value.
  It means that the converter is set into RAW_MODE mode, with conversion server set to port 5000 investigation and baud rate of 9600 with no parity. When parameters are different than default ones and we want to restore them, we can establish it by pressing and holding reset button. When procedure of restoring the settings has been initiated, the COTER-E would announce it to us by lighting status (red) diode.
- Send "SetCoterConfig" command with first byte in parameter field set to value 2. To change communication protocol only, there is no required sending all parameters, it is enough to send one byte only with value 2 (NVT_MODE mode) in parameter field. When there are no other parameters, the command will be accepted (detailed information on this subject see chapter Converion server configuration).
- After having sent the command, it is possible to verify the parameter values once more by sending "GetCoterConfig" command and checking if turned parameters meet ones which were expected.

Configuration flow-chart is showed in picture 3.

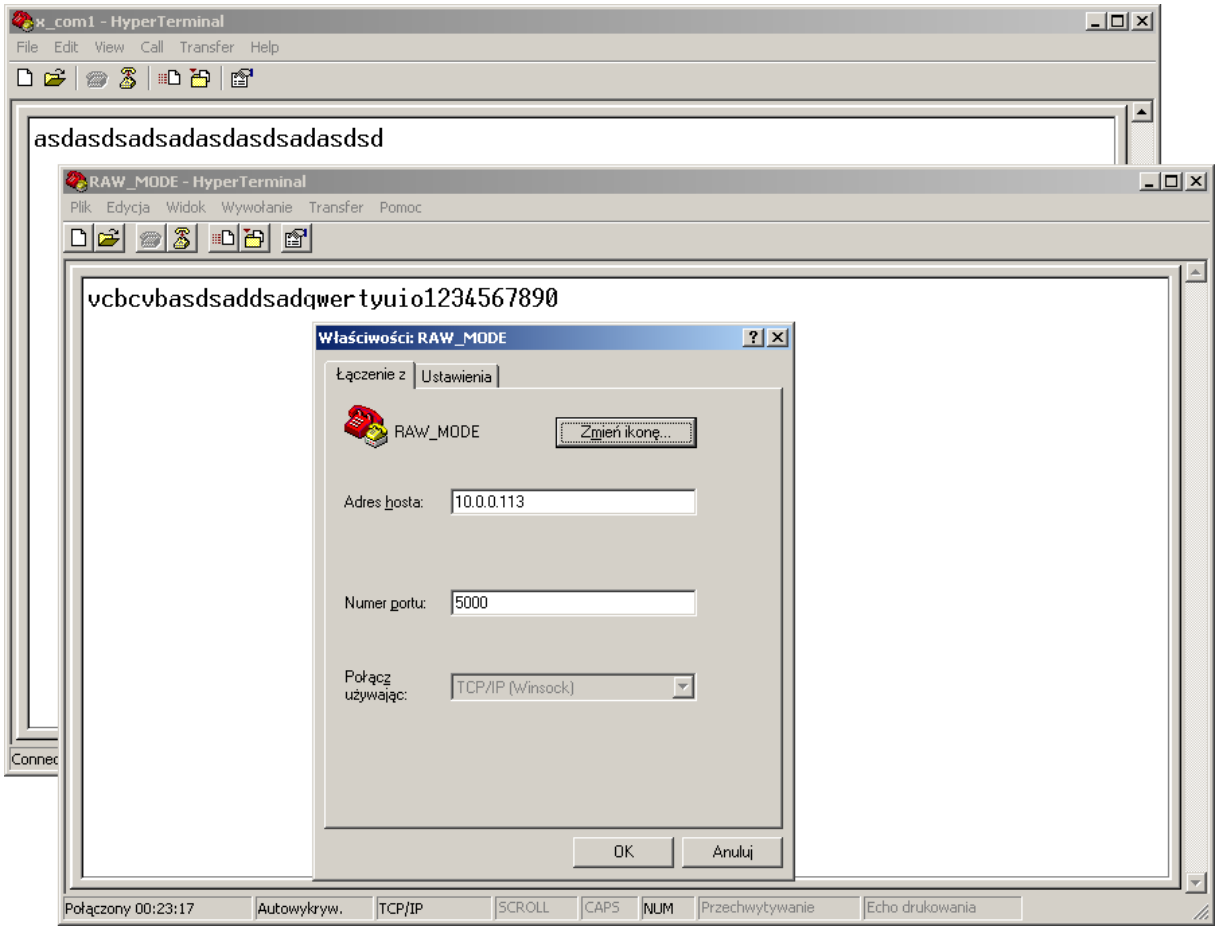


Picture 3: Changing the converter mode for NVT_MODE

6.3. Connection with conversion server

6.3.1. RAW_MODE

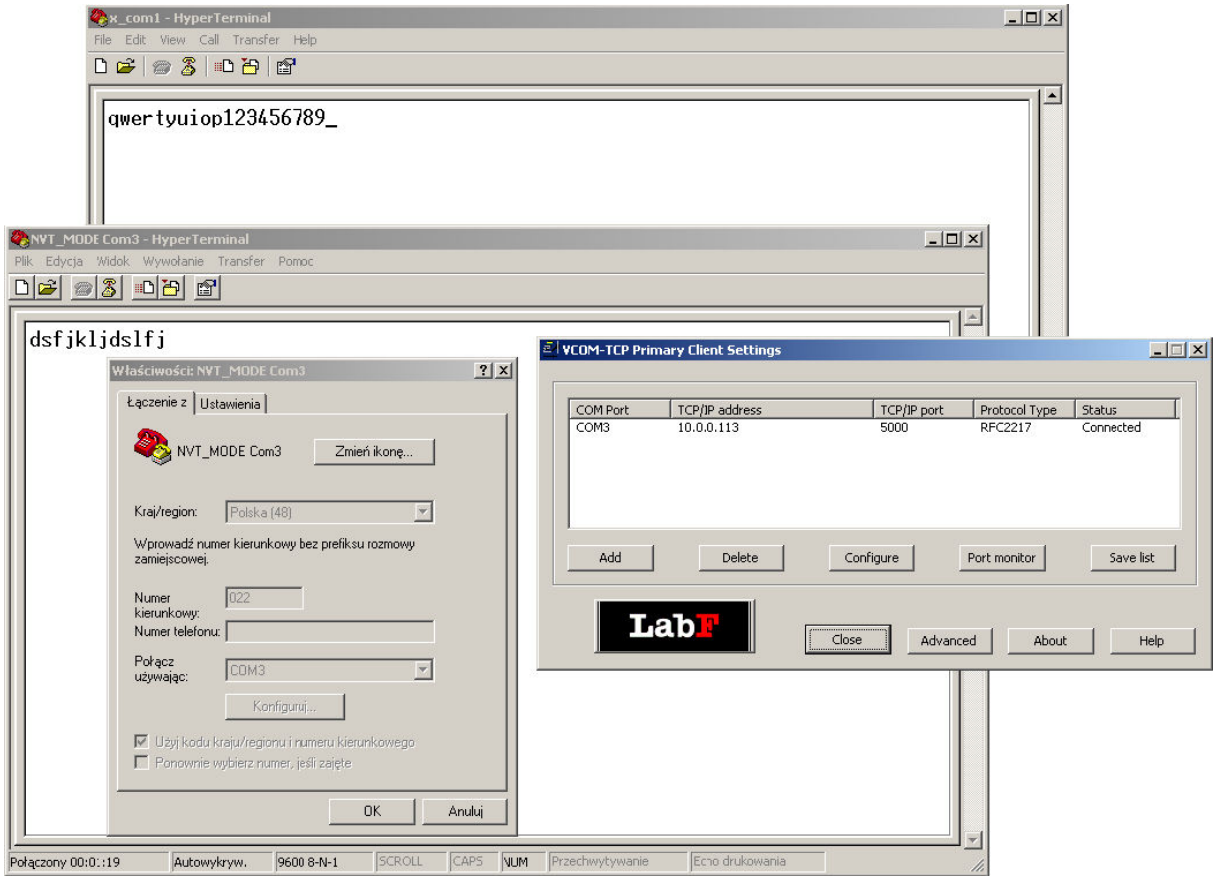
In RAW_MODE the converter sends data directly from Ethernet TCP/IP port (by defaults 5000) of conversion server to serial interface port, with making no changes. Baud rate of serial port is set by means of configuration server (port 50). By defaults, serial interface is set to: baud rate 9600 bps, 8 bits per frame, no parity and disabled flow control. In this mode control the flow is possible only by using automatic flow control mode. To establish connection, you can use "Hyper Terminal" program. Set connection flow for TCP/IP (Winsock), write proper port and IP address in, and confirm the selections. To verify correctness of converter operation, connect it to the same PC which established connection with converter recently, and observe correctness of the transmission.



Picture 4: Connection established in RAW_MODE

6.3.2. NVT_MODE

In NVT_MODE, one must use additional software which multiplies/removes presence of 0xFF in flow stream (depending on flow direction). Commands which control serial port parameters and data flow are sent via the same communication port complying with "Com Port Control Option" standard described in RFC-2217 which has been illustrated shortly in chapter NVT protocol description. In the same mode, you can use an application which creates virtual serial port, owing to it the converter can act as a virtual serial port available in list of physical ports installed on PC. An example of such software is "VCOM-TCP Primary Client" module from datAxe-v2.1 pack. In picture 5 are showed programs which are used for establishing the connection with converter. This mode allows changing the baud rate, controlling the RTS/CTS (DTR/DSR) lines and as a standard extension controlling the I/O port, with no need of establishing a new connection.



Picture 5: Connection in NVT_MODE using virtual port

7. Data transmission format using configuration server

In this technical data sheet, Ethernet protocol description has been confined to explanation of commands, responses and their parameters. Header and control sum CRC exist always and comply with full technical data sheet of NETRONIX protocol.

Command frame:

header	C_CommandName	Command_parameters1…n	CRC

Response frame:

header	C_NazwaRozkazu +1	Response_parameters1…m	OperationCode	CRC

Operation of Netronix protocol can be tested by means of free FRAMER software.

8. Converter configuration

It is possibly to control the converter in any way by means of configuration server. In case of faulty settings, there is possibility of restoring the defaults. To achieve this, press user button, using small wire and hold this button pressed for approximately 5 seconds. After initiating, status LED (red) will light immediately, and will fade away then, after stopping the operation.

8.1. Network configuration

Commands described in this section are connected with network configuration. By means of these commands can modify the settings or check their current status.

8.1.1. Physical address changing

By defaults, MAC physical address is "00-04-A3-00-00-00". In case of address discrepancy with other devices, it is possible to modify them.

Command frame:

header	C_SetMAC	MAC	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_SetMAC	1	MAC physical address change	0x90
MAC	6	New MAC address	New MAC address 00 00 00 00 00 00 to FF FF FF FF FF FF

Response frame:

header	C_SetMAC +1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

The command sets flag of announcement on setting update necessity. Just after execution of "Reset" command, new assigned parameter is used for communication in network.

8.1.2. Reading the physical address MAC

Command frame:

header	C_GetMAC		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_GetMAC	1	MAC physical address readout	0x92

Response frame:

header	C_GetMAC +1	MAC	OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
MAC	6	MAC physical address	
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0xFF – OC_Successful – command executed successfully

Returns MAC address of converter. If its address has been changed and settings have not been updated up to that time, is returns an old address value with which the converter operates currently.

8.1.3. Setting the network parameters

By defaults, the DHCP service is active.

Command frame:

header	C_SetInterfaceConfig	DHCP	AdrIP	NetMask	GateWay	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_SetInterfaceConfig	1	Changing the network parameters	0x54
DHCP	1	Flag of enabling the DHCP client	0 –DHCP client enabled 1 – DHCP client disabled
AdrIP(1)	4	New address of network	Range of IP address altering
NetMask(1)	4	New mask of network	Range of IP address altering
GateWay(1)	4	New address of network gate	Range of IP address altering

Notes:

(1) When DHCP client activation flag is set to 1; other parameters AdrIP, NetMask, GateWay are optional, because these data are rewritten by configuration made from server. Otherwise, all data must be present during sending the command. Optional data will be used by the converter, when the DHCP server is non-active.

Response frame:

header	C_SetInterfaceConfig + 1		Operation code	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or incorrect their values 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

The command sets flag of announcement on setting update necessity. Just after execution of "Reset" command, new assigned parameter is used for communication in network.

8.1.4. Reading out the network parameters

Command frame:

header	C_GetInterfaceConfig		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_GetInterface Config	1	Reading out the network parameters	0x56

Response frame:

header	C_GetInterface Config +1	DHCP	AdrIP	NetMask	GateWay	OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
DHCP	1	Flag of switching DHCP client on	0 – DHCP client enabled 1 –DHCP client disabled
AdrIP(1)	4	New address of network	Range of IP address altering
NetMask(1)	4	Current mask of subnetwork	Range of IP address altering
GateWay(1)	4	Current address of gate	Range of IP address altering
Operation code(1)	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0xFF – OC_Successful – command executed successfully

Notes:
(1) When flag of DHCP client activation is set to 1; other parameters which are returned (AdrIP, NetMask, GateWay), are the same as they have been assigned by DCHP server.

Returns network parameters of converter. If its address has been changed and settings have not been updated up to that time, it returns an old address value with witch the converter operates at this time.

8.1.5. Changing the network name

In case of default configuration, the network name is "COTER-EXXX-VY", where sequence "XXX" is device version with its details on its features, but “Y” is the number describing the version of firmware saved.

Command frame:

header	C_SetNetBiosName	NewName	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_SetNetBiosName	1	Network parameters change	0x94
NewName	16 max.	New name of network	Maximum 15 ASCII dots with ending of 0x00 value

Response frame:

header	C_SetNetBiosName + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters

			0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

The command sets flag of announcement informing on setting update necessity. Just after execution of "Reset" command, new assigned parameter is used for communication in network.

8.1.6. Reading out the network name

Command frame:

header	C_GetNetBiosName		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_GetNetBiosName	1	Readout of network setting parameters	0x96

Response frame:

header	C_GetNetBiosName +1	NetName	OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
NetName	16 max.	Network name	The name finished 0x00
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0xFF – OC_Successful – command executed successfully

Returns network name of converter. If its address has been changed and settings have not been updated up to that time, it is returns an old address value with which the converter has operated till now.

8.2. Controlling the access

The converter was protected against access of intruders, based on verification valid IP addresses from which one can establish a connection. Maximal value of addresses which can be entered is 4. Not to make changes by intruders, the converter is equipped with login password to commands which modify its settings. If password is set (is not blank), a user can read all settings excluding ACL list. Readout of ACL list is protected as well. By default configuration, password is blank, i.e. all settings can be made with not need of login. When given password is active, and a user will send proper command with valid password, the converter begins to count the time i.e. time which expires since last command has been received. After crossing the 60 seconds since last command, a user will be log-out automatically.

8.2.1. Changing the settings of ACL list

By default configuration, ACL list check is disabled. If a computer with IP address which is not on an ACL list (which is not blank) is trying to connect with a converter, this action is stopped immediately by the converter, after establishing the connection.

Command frame:

header	C_SetAccesControllList	AdrIP_1	AdrIP_2	AdrIP_3	AdrIP_4	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_SetAccesControllList	1	Change of access list addresses	0x5C
AdrIP_1	4	IP address of master host 1	Range of IP address altering
AdrIP_2	4	IP address of master host 2	Range of IP address altering
AdrIP_3	4	IP address of master host 3	Range of IP address altering
AdrIP_4	4	IP address of master host 4	Range of IP address altering

Response frame:

header	C_SetAccesControllList + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

The command sets flag of announcement on setting update necessity. Just after execution of "Reset" command, new assigned parameter is used for communication in network.

8.2.2. Reading out the ACL list

Command frame:

header	C_GetAccesControllList		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_GetAccesControllList	1	ACL access list read-out	0x5E

Response frame:

header	C_GetAccesControllList +1	AdrIP_1	AdrIP_2	AdrIP_3	AdrIP_4	OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
AdrIP_1	4	IP address of master host 1	Address 1 from which can establish connection
AdrIP_2	4	IP address of master host 2	Address 2 from which can establish connection
AdrIP_3	4	IP address of master host 3	Address 3 from which can establish connection
AdrIP_4	4	IP address of master host 4	Address 4 from which can establish connection
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

Returns IP address list from which the connection has been established. If the list has been changed before, and converter settings have not been updated, it returns an old list the converter uses at this time.

8.2.3. Login

Command frame:

header	C_LoginUser	Password	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_LoginUser	1	Start of session	0xB2
Password	9 max.	Login password	Login password with 0x00 ending

Response frame:

header	C_LoginUser + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0x09 - OC_WrongPassword – wrong password 0xFF - OC_Successful – command executed successfully

Sending the command with valid password, allows user to modify parameters of converter operation. If password service is disabled, sending the "LoginUser" command with any chosen password will cause sending the return message with information that the operation has finished successfully.

8.2.4. Log-out

Command frame:

header	C_LogoutUser		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_LogoutUser	1	End of session	0xD6

Response frame:

header	C_LogoutUser + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0xFF – OC_Successful – command executed successfully

If password which is saved in set-up memory is not 0, user will be log-out.

8.2.5. Changing the password

Command frame:

header	C_ChangeLoginUser	NewPassword	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_ChangeLogin User	1	Login password change	0xB4
NewPassword	9 max.	New password	8 bytes with 0x00 ending max.

Response frame:

header	C_ChangeLoginUser + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters 0x09 – OC_WrongPassword – no user

			login 0xFF – OC_Successful – command executed successfully

Changes existing login password. If new password is blank (0x00 only), the login service will be disabled.

8.3. Configuring the conversion server

8.3.1. Changing the settings of conversion server

Communication parameters of serial interface can be modified in two ways. One way is operation of configuration server, which modifies baud rate, and also makes data to write into nonvolatile memory. Owing to this, on the next start of the computer it performs set-up of parameters assigned lately by means of configuration server.

Command frame:

header	C_SetCoterConfig	Mode	Port	Bitrate	Parity	DirFlowControl	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_SetCoterConfig	1	Write of conversion server settings	0xA0
Mode	1	Mode of conversion server	0x00 – OFF_MODE 0x01 – RAW_MODE 0x02 – NVT_MODE
Port	2	Communication port of conversion server	[LSB MSB] 1024 to 65535
Bitrate	4	Baud rate	[LLSB LMSB HLSB HMSB] 1200 to 230400 bps
Parity	1	Parity bit	0x01 – No parity (None) 0x02 – Odd (Odd) 0x03 – Even (Even) 0x04 – One always (Mark) 0x05 – Zero always (Space)
DirFlowControl	1	Flow control direction	0x00 – Flow control using RTS/CTS line 0x01(1) – Flow control using DTR/DSR line 0x02(1) (2) – Automatic flow control using RTS/CTS line

Response frame:

header	C_C_SetCoterConfig + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning

Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

Notes:
(1) Option is available only with device COTER-E2.
(2) Option is not available in case of NVT_MODE.

In case of "SetCoterConfig" command, not all parameters are needed, some of them can be omitted, but sequence should be fixed. It means that if we want to change parity bit, we should give former parameters as well. When modifying the conversion port, "NeedUpdateSettings" flag changes its state to active which means that acceptance of change needs performing the "Reset" command at first. When converter operates in NVT_MODE with TCP connection active, a change of serial interface speed and parity bit will not be made immediately. The reason is that it could disturb the operation of remote host, which controls flow speed by means of NVT commands. The change will be made during next establishing of connection only.

8.3.2. Reading out the settings of conversion server

Command frame:

header	C_GetCoterConfig		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_GetCoterConfig	1	Reading out the settings of conversion server	0xA2

Response frame:

header	C_GetCoter Config + 1	Mode	Port	Bitrate	Parity	DirFlow Control	Operatio nCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Mode	1	Mode of conversion server	0x00 – OFF_MODE 0x01 – RAW_MODE 0x02 – NVT_MODE
Port	2	Communication port of conversion server	[LSB MSB] 1024 to 65535
Bitrate	4	Baud rate	[LLSB LMSB HLSB HMSB] 1200 to 115200 bps
Parity	1	Parity bit	0x01 – no parity bit (None) 0x02 – odd (Odd) 0x03 – even (Even) 0x04 – one always (Mark)

			0x05 – zero always (Space)
DirFlowControl	1	Flow direction	0x00 – flow control using RTS/CTS line 0x01 – flow control using DTR/DSR line
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0xFF – OC_Successful – command executed successfully

8.4. Controlling the I/O ports

Commands described in this chapter relate to COTER-E2xx version of the converter. Converters of other type do not have I/O ports.

8.4.1. Saving the settings of I/O ports

We use "SetIOConfig" command to configure the I/O ports. Using this command one can change direction of port operation, modify port status and save initial configuration for the device during power-up.

Command frame:

header	C_SetIOConfig	IONum	IODir	IOVal	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_SetIOConfig	1	I/O port configuration	0x50
IONum	1	I/O port number	Value range 0 to 1
IODir	1	Operation direction of the port	0 – Port configured as an output 1 – Port configured as an input
IOVal	1	Initial value of the port	Value range 0 to 1

If port is configured as an input, initial status will not be required. If port is configured as an output, its current status will be replaced with value given in IOVal parameter.

Response frame:

header	C_SetIOConfig + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

8.4.2. Reading out the I/O settings

Command frame:

header	C_GetIOConfig	IONum	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_GetIOConfig	1	Reading out the configuration parameters of I/O ports	0x52
IONum	1	I/O port number	Value range 0 to 1

Response frame:

header	C_GetIOConfig + 1	IODri	IOVal	OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
IODir	1	Operation direction of the port	0 – Port configured as an output 1 – Port configured as an input
IOVal	1	Initial value of the port	Value range 0 to 1
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0xFF – OC_Successful – command executed successfully

8.4.3. Changing the status of an output port

Command frame:

header	C_WriteOutputs	PortNum	PortVal	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_WriteOutputs	1	Status change of port configured as an output	0x70
PortNum	1	Port number	Port number 0 to 1.
PortVal	1	New status of the port	Value not equal zero interpreted as a high state.

Response frame:

header	C_WriteOutputs + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation	1	Status of command execution	0x04 – OC_ParameterError – wrong

code			number of parameters or their range 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

If port has been configured as an input, the effect of the command would have been returning operation code "ParameterError".

8.4.4. Reading out the input status

Command frame:

header	C_ReadInputs	IONum	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_ReadInputs	1	Reading out the port status	0x72
IONum	1	Port number	Port number od 0 to 1.

Response frame:

header	C_ReadInputs + 1	IOVal	OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
IOVal	1	Current status of the port	Value range od 0 do 1
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0xFF – OC_Successful – command executed successfully

8.5. General purpose commands

8.5.1. Inquiry on status

The converter has a flag informing on its unique state. The flag informs user that correct frame of setting modification related to network interface is received. Change of settings requires reset of the device (using command “Reset”), to its network parameters which would be accepted and used in communication.

Command frame:

header	C_GetNeedUpdateFlag		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_GetNeedUpdateFlag	1	Readout of status flag	0xD4

Response frame:

header	C_GetNeedUpdateFlag + 1	NeedUpdateFlag	OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
NeedUpdate Flag	1	Flag modified for network purposes	0x00 – Device operates with current settings 0x01 – Device needs update of settings
Operation code	1	Status of commands execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0xFF – OC_Successful – command executed successfully

8.5.2. Reset command

The command makes converter to reset. After reset, all modified parameters are red-out from nonvolatile memory and used in network communication.

Command frame:

header	C_Reset		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_Reset	1	Device reset	0xD0

Response frame:

header	C_Reset + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

8.5.3. Software version

The command returns software version and name of the device.

Command frame:

header	C_FirmwareVersion		CRC

Wherein:

Parameter name	Length	Parameter	Value range

	[byte]	description	
C_FirmwareVersion	1	Software version	0xFE

Response frame:

header	C_FirmwareVersion + 1	Firmware	OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Firmware	Volume depending on length of software name	Software version	ASCII sequence in which is written name and software number
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0xFF – OC_Successful – command executed successfully

8.5.4. Bootloader activation command

The command makes converter to reset. After reset, the device begins to operate with last used parameters of network interface and makes TFTP server to operate. The server is active for 30 seconds since start. If during this time loading procedure of firmware is not initiated, the device will go back to performing of main program. If from any reason loading of firmware is broken (e.g. supply decay), the device will remain in bootloader mode, until loading will come to the end successfully.

Command frame:

header	C_RunBootloader		CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Value range
C_RunBootloader	1	Run of bootloader	0xD2

Response frame:

header	C_RunBootloader + 1		OperationCode	CRC

Wherein:

Parameter name	Length [byte]	Parameter description	Meaning
Operation code	1	Status of command execution	0x04 – OC_ParameterError – wrong number of parameters or their range 0x09 – OC_WrongPassword – no user login 0xFF – OC_Successful – command executed successfully

9. NVT protocol description

COTER-E2I and COTER-E4I converters can operate in two modes of data conversion Ethernet <-> Serial Port. In RAW_MODE, the converter operates as a device, which sends data directly via TCP/IP port (by defaults 5000) to a serial port. Modification of transmission parameters is performed via configuration server operating on port 50. In the configuration, it is not possible to control data flow by means of RTS/CTS and DSR/DTR signals.

NVT_MODE of the server is based on standard described in RFC-2217, which is option of TELNET protocol described in RFC-854. Owing to used extended protocol "Telnet Com Port Control Option", one can perform modifications of flow parameters, check of error status or control of serial interface data flow, using the same connection which is used for data transmission. The server is designed to be compatible with conversion programs of serial data to TCP/IP data stream. By means of this type programs, the converter can be added as a new virtual serial port, which can be used in the same way as in case of serial port installed in a PC computer. To optimize compatibility with RFC-2217 standard, the server has been designed and tested using datAxev2.1 program. The protocol mentioned before has been improved with additional functionality allowing to control of I/O ports remotely. Using software of virtual serial port, it is not possible to change I/O port status of the converter, when communications with conversion sever are established. In such case is advised to maintain second connection with conversion server. Owing to this, it possible to change any I/O port settings, not disturbing operation of the converter. When converter operates in NVT_MODE, any change of settings using configuration server has not any influence on current settings of port but on initial conditions during succeeding connection with server. Parameters sent with control frame are saved in nonvolatile memory of the converter and set during following start of supply. Owing to this, it is possible to perform single configuration, and subsequent start-ups will recover last setting used.

9.1. Short description of protocol

Because there is no need of full implementation, not all commands of protocols mentioned before has been implemented. In next subchapters we will mention and describe shortly some commands used by converter. To get more familiar with the protocol, see their documentation.

9.1.1. Commands implemented from RFC-854 standard

NVT commands are transmitted in the same TCP/IP stream as data for conversion. Each NVT command is preceded with prefix IAC (0xFF). Some of the commands such as NOP and AYT are of two-byte length. More complex commands must be surrounded with markers which are built of two dots: start <IAC><SB> and stop <IAC><SE>. Data within markers are interpreted as one command. Each appearance of 0xFF, which is to be interpreted as data byte or parameter, but not as IAC mark, must be repeated two times. When server operating in NVT_MODE, receives byte 0xFF from serial interface, it will duplicate the byte and transmit data via TCP/IP port. In RAW_MODE, NVT command processor is disabled, so there is no need to duplicate appearance of 0xFF.

Table 2: Supported commands from RFC-854 standard

HEX value	Mnemonic	Description
0xF0	SE	Indicates end of option and their parameters.

0xF1	NOP	No operation.
0xFA	SB	Indicates beginning of option and their parameters.
0xFB	WILL	Shows willingness to begin or confirmation that indicated option is being executing.
0xFC	WON'T	Shows willingness to stop execution or refuses to execute indicated option.
0xFD	DO	Indicates execution or expects indicated option to execute.
0xFE	DON'T	Indicates stopping the execution or shows that indicated option is not being executing.
0xFF	IAC	Interpret it as a command – If following data byte is not equal 0xFF, it will be treated as a command.

9.1.2. Commands interpreted from RFC-2217 protocol standard

RFC-2217 protocol describes method of remote asynchronous control of serial port using TCP/IP connection. By means of this protocol, it is possible to change baud rate, parity, flow control or verify transmission errors. In description of "Telnet Com Port Control Option" (0x2C) there are two kinds of mnemonic: CAS determining the client -> server query, and ASC determining the server -> client response. Relation between mnemonics is ASC = CAS + 0x64 (100 DEC).

Table 3: Using "Com Port Option" from RFC-2217 protocol

DEC value	Mnemonic	Description
0	CAS_SIGNATURE	Exchanging the information on signature.
1	CAS_BAUDRATE	Baud rate (1200 to 115200 bps).
2	CAS_DATASIZE	Data frame length (possibility to send frames of 8 bits length).
3	CAS_PARITY	Data parity.
4	CAS_STOPSIZE	Number of stop bits (limited to 1 bit).
5	CAS_CONTROL	Data flow control.
6	CAS_LINESTATE	Serial interface status.
7	CAS_MODEMSTATE	Status of lines which drive control flow.
10	CAS_SET_LINESTATE_MASK	Memo mask of serial interface status.
11	CAS_SET_MODEMSTATE_MASK	Memo mask of drive line status.
12	CAS_PURGE_DATA	Cleaning the I/O buffers.
0+100	ACS_SIGNATURE	Server response with signature information.

1+100	ASC_BAUDRATE	Server response with baud rate.
2+100	...	...
...	...	...

9.1.3. Controlling the lines of I/O ports

To control I/O lines by means of TCP/IP port used for data transmission, the RFC-2217 protocol has been provided with extra options.

Table 4: Extension of RFC-2217 protocol for controlling I/O ports

DEC value	Mnemonic	Description
50	CAS_SET_GPIO	Checking and controlling the status of I/O lines.
51	CAS_SET_REG_GPIO	Controlling many lines at once by means of I/O port register.
50+100	ASC_SET_GPIO	Response to check or I/O line status change.
51+100	ASC_SET_REG_GPIO	Response to change of register controlling the I/O lines.

Note:
Extended commands of the protocol concern to devices with I/O ports only e.g. COTER-E21.

For controlling single I/O lines, CAS_SET_GPIO (50) option with parameter sent is used.

Table 5: Parameters of CAS_SET_GPIO option

Parameter of CAS_SET_GPIO (HEX) option	Meaning
0x00	Returns register of I/O ports, no matter if the port is set as an input or as an output.
0x10 ... 0x11	Set high state of x line. Number of line is calculated as a logic product of parameter and 0x0F. If port is set as an input, the command will be ignored.
0x20 ... 0x21	Set low state of x line. Number of line is calculated as a logic product of parameter and 0x0F. If port is set as an input, the command will be ignored.

Note:
As a response to query 0x10...0x11 and 0x20...0x21, state of I/O port control register will be returned being the response to 0x00 query at the same time. Owing to the response returned, a correctness of I/O port state can be verified. If returned value differs from value set before, a configuration of port should be checked, independently if the port has been set as an output or not.

For controlling multiple I/O lines, the CAS_SET_REG_GPIO (51) option with parameter sent is used.

Table 6: Parameters of CAS_SET_REG_GPIO option

Parameter of CAS_SET_GPIO (HEX) option	Meaning
0x00 ... 0x03	Sets all outputs in a state set by bits which are in suitable positions. As a response, it returns I/O port register, no matter if the port is set as an input or output.

9.2. Example of communication between client and server

In this chapter, exemplary queries and responses will be described, which can take part in communication between client and server.

9.2.1. Negotiating the service of „Telnet Com Port Control Option”

During attempt to establish the connection with converter which operates in NVT_MODE, the server sends an await message for service of "Telnet Com Port Control" option and waits for answer which confirms command service by client. If client does not confirm the option service, the server will switch to RAW_MODE automatically. On following connection, the server will still wait for the client who services NVT commands.

Query of a server:
<IAC><WILL><COM_PORT_OPTION>
<0xFF><0xFB><0x2C>

Response of a client:
<IAC><WILL><COM_PORT_OPTION>
<0xFF><0xFD><0x2C>

9.2.2. Querying on device signature

Result of execution the command by the converter is returning the dot sequence informing user on device on device and firmware version.

Query of a server:
<IAC><SB><COM_PORT_OPTION><CAS_SIGNATURE><IAC><SE>
<0xFF><0xFA><0x2C><0x00><0xFF><0xF0>

Response of a client:
<IAC><SB><COM_PORT_OPTION><ASC_SIGNATURE><IAC><SE>
<0xFF><0xFA><0x2C><0x64><tekst><0xFF><0xF0>

9.2.3. Changing the baud rate or serial interface

To accept the command and change the baud rate, there must be present in the field <parametr> a value from range 1200 to 115200, written as 32-bit integer value. If in query in field <parametr> there are 0x00’s only or value which is outside this range, the server will respond with current baud rate value.

Query of a server:
<IAC><SB><COM_PORT_OPTION><CAS_BAUDRATE><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x01><0x00 0x00 0xE1 0x00><0xFF><0xF0>

Response of a client:
<IAC><SB><COM_PORT_OPTION><ASC_BAUDRATE><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x65><0x00 0x00 0xE1 0x00><0xFF><0xF0>

As a result of query is change of serial interface baud rate to 57600 bps.
<0x00 0x00 0xE1 0x00> = 57600bps.

9.2.4. Changing the size of serial interface data frame

If in query in field <parametr> there is the value of 0x00, the server will respond with current data frame size, located in parameter of server response.

Query of a server:
<IAC><SB><COM_PORT_OPTION><CAS_DATASIZE><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x02><0x00><0xFF><0xF0>

Response of a client:
<IAC><SB><COM_PORT_OPTION><ASC_DATASIZE><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x66><0x08><0xFF><0xF0>

The converter can operate with 8-bit data frames only. Attempt of changing the frame size will make it to send response with current frame size.

9.2.5. Changing the parity bit of serial interface

If in query in field <parametr> there is the value of 0x00, the server will respond with current parity mode in which at this time converter operates.

Query of a server:
<IAC><SB><COM_PORT_OPTION><CAS_PARITY><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x03><0x00><0xFF><0xF0>

Response of a client:
<IAC><SB><COM_PORT_OPTION><ASC_PARITY><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x67><0x01><0xFF><0xF0>

Available values of <parametr>
- 0 – query for current mode;
- 1 – no parity bit;
- 2 – odd;
- 3 – even;
- 4 – sign;
- 5 – space.

9.2.6. Changing the flow control, changing the RTS and DTR line status

If in query in field <parametr> there is the value of 0x00, server will respond with current mode of flow control in which at this time converter operates.

Query of a server:
<IAC><SB><COM_PORT_OPTION><CAS_CONTROL><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x05><0x01><0xFF><0xF0>

Response of a client:
<IAC><SB><COM_PORT_OPTION><ASC_CONTROL><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x69><0x01><0xFF><0xF0>

Available values of <parametr>
- 0 – query for current flow control mode;
- 1 – sets no flow control;
- 3 – hardware flow control;
- 7 – query for DTR line status;
- 8 – sets DTR signal in active state;
- 9 – sets DTR signal in non-active state;
- 10 – query for RTS line status;
- 11 – sets RTS signal in active state;
- 12 – sets RTS signal in non-active state.

9.2.7. Changing the status of I/O port

If in query in field <parametr> there is the value of 0x00, the server will respond with current status of register controlling the I/O ports. No matter how the port is configured, returned value will represent logic state present on output or input.

Query of a server:
<IAC><SB><COM_PORT_OPTION><CAS_SET_GPIO><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x32><0x10><0xFF><0xF0>

Response of a client:
<IAC><SB><COM_PORT_OPTION><ASC_SET_GPIO><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x96><0x01><0xFF><0xF0>

If port 0 is configured as output, as a result of execution of the query will be a change of output state of the port to high state.

Query of a server:
<IAC><SB><COM_PORT_OPTION><CAS_SET_GPIO><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x32><0x21><0xFF><0xF0>

Response of a client:
<IAC><SB><COM_PORT_OPTION><ASC_SET_GPIO><parametr><IAC><SE>
<0xFF><0xFA><0x2C><0x96><0x02><0xFF><0xF0>

If port 1 is configured as output, result of execution of query will be a change of output state of the port to low state. In this case, the server returns information in parameter that in spite of

sending the command, the port is still in high state. The reason of it can be setting the port as an output.

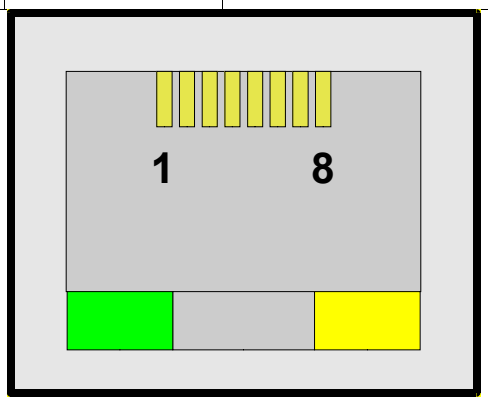
10. Description of signal on connectors

Pictures below show connectors from their front sides.

Connector Ethernet

Table 7: Signals on Ethernet connector

Pin	Nazwa	Opis
1	TP_TX+	Pair of TX (+)
2	TP_TX-	Pair of TX (-)
3	TP_RX+	Pair of RX (+)
4	-	Non-connected
5	-	Non-connected
6	TP_RX-	Pair of RX (-)
7	-	Non-connected
8	-	Non-connected



Picture 6: Ethernet connector

Table 8: LED's in Ethernet connector

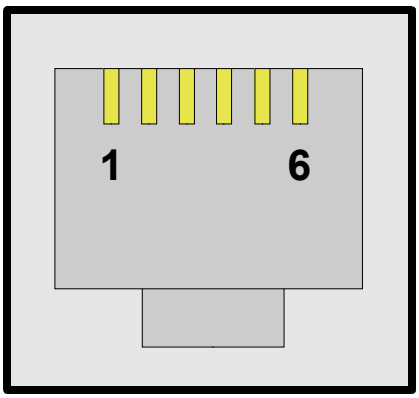
LED	Description
Yellow (on the right)	Ethernet 10 activity
Green (on the left)	Ethernet link

Connector RJ-12

Concerns to COTER-E4I.

Table 9: Signals in RS-485, supply connector

Pin	Name	Description
1	VCC	Supply voltage: +8 to +16 V DC, 300 mA max.
2	GND	Supply ground
3	A	“A” (+) wire for RS-485
4	B	“B” (-) wire for RS-485
5	-	Non-connected
6	-	Non-connected



Picture 7: RS-458, supply connector

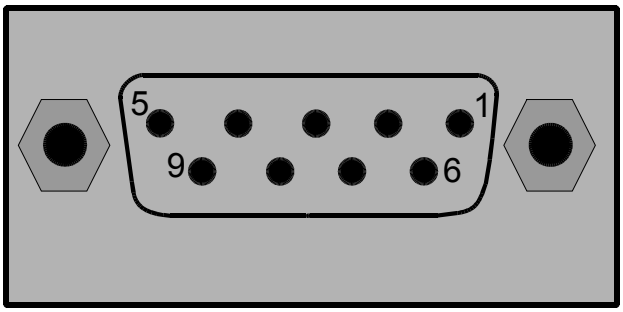
Connector DB-9

Concerns to COTER-E2I.

Table 10: Signals in RS-232, I/O, supply connector

Pin	Name	Description
1	-	Non-connected
2	RXD (RS-232)	Data received (input).
3	TXD (RS-232)	Data sent (output).
4	VCC	Supply voltage: +8 to +16 V DC, 300 mA max.
5	GND	Supply ground.
6	IO_0	Port no. 0 (input/output).
7	RTS/DTR (RS232)	Ready to send the data/device ready (output).

Pin	Name	Description
8	CTS/DSR (RS232)	Ready to receive the data/data set ready (input).
9	IO_1	Port no. 1 (input/output).



Picture 8: RS-232, I/O, supply connector

11. Electrical specifications

Table 11: Electrical parameters

No.	Symbol	Characteristics	Value			Unit	Notes
			Min.	Typically	Max.		
Supply parameters							
1	Vpower (1)	Supply voltage	+8.0	+12.0	+16.0	V	-
2	Ipower (2)	Supply current	110	130	300	mA	-
RS-232 port parameters							
3	Vrs232_out (3)	Range of output voltage fluctuations on TXD/RTS(DTR) lines	-	±10.0	-	V	With no load
4	Vrs232_in (3)	Maximum range of input voltage fluctuations on RXD/CTS(DSR) lines	-	-	±30.0	V	-
I/O port parameters							
5	Vio_high_out (3)	Output voltage of I/O port in high state	3.2	3.3	3.4	V	With no load
6	Vio_low_out (3)	Input voltage of I/O port in low state	0	0.05	0.10	V	With no load
7	Vio_high_in (3)	Input voltage of I/O port in high state	2.4	3.3	5.5	V	-
8	Vio_low_in (3)	Input voltage of I/O port in low state	-0.3	0.0	0.8	V	-
9	Rio_out (3)	Output resistance of I/O port	78	82	86	Ω	-
RS-485/422 port parameters							
7	Vrs485_out (4)	Range of differential output voltage abs(A, B)	1.5	2.7	5.0	V	-
8	Vrs485_in (4)	Range of differential input voltage A, B	-	-	±14	V	-

Notes:

(1) Supply voltage should be stabilized.

(2) Supply current depends on supply voltage. For $V_{power} = V_{max}$ -> $I_{power} = I_{min}$.

(3) For COTER-E2I version.

(4) For COTER-E4I version.