



# Highcross NetString

## Data Exchange Protocol (revision 2.20)

The data exchange between a main controller and Highcross devices occurs as Telnet-like plain text messages via TCP/IP. Each command, request or message has the termination chars CR+LF (0x0D and 0x0A) and is sensitive to space chars.

### message **DEVICE**

The message **DEVICE** is sent by every device to a controller after the connection is established.

Syntax: **DEVICE**<space>=<space><device name>

Example: *DEVICE = IO16*

### request and message **PING PING\_REPLY**

After receiving the request **PING** from a controller every device sends the message **PING\_REPLY** if it is active

### messages **PUSH-RELEASE PUSHED-RELEASED**

The pair messages **PUSH** and **RELEASE** are sent by a device when a digital input is activated.

(**PUSH** – is an event of closure of **Normally Opened** input or opening of **Normally Closed** input, **RELEASE** is an opposite event).

Syntax: **PUSH**<input number>, **RELEASE**<input number>

Example: *PUSH[9]  
RELEASE[9]*

The pair messages **PUSHED-RELEASED**, in turn, specify the current state of inputs and are sent automatically after connecting to the controller or as a reply to request “?”

Syntax: **PUSHED**<input number>, **RELEASED**<input number>

Example: *PUSHED[9]  
RELEASED[9]*

### commands **ON – OFF**

Used to turn digital outputs of devices ON and OFF

Syntax: **ON**<output number>, **OFF**<output number >

Example: *ON[9]  
OFF[9]*

Note: The ECM-IO16 module will send also PUSH and RELEASE messages, since inputs and outputs are physically the same.

After disconnection from a controller a device will switch off all outputs that were “ON” at that moment, if the option “Remember the state of digital output” was not checked in the configuration menu.

### Command **PULSE**

Used to turn on the digital output temporarily.

Syntax: **PULSE**<output number> (default time is 0.5 sec).  
**PULSE**<output number>**T**<time> (time is set in tenths of seconds)

Example: *PULSE[9]T15*  
(Turn on the output number 9 for 1.5 seconds)

If the output is turned on at the moment of command, it will turn off after a set time.

### command **INV**

Used to invert the state of digital output

Example: *INV[9]*

### request **?**

Used to get the current state of the device.

Syntax: **?**

**?<modifier>** where modifier can be “IN”, “OUT” or “ALL”  
**?<modifier>**<number>

Examples: *? ?ALL ?IN ?IN[9] ?OUT ?OUT[9]*

### Error messages

The following error messages are sent by devices as an answer to incorrect commands:

**ERR\_UNKNOWN\_COMMAND**  
**ERR\_INCORRECT\_COMMAND**  
**ERR\_ILLEGAL\_PORT**  
**ERR\_ILLEGAL\_OUTPUT**

**ERR\_ILLEGAL\_INPUT**  
**ERR\_ILLEGAL\_LEVEL**  
**ERR\_INPUT\_BUFFER\_OVERFLOW**  
**ERR\_OUTPUT\_BUFFER\_OVERFLOW**

Other device-specific commands and messages are described in the following sections.

<b>EPM-DM3D</b>	3 dimmer outputs (220V, adjustment range 0-255) 6 digital inputs
<b>ECM-DM6D</b>	6 dimmer outputs (220V, adjustment range 0-255) 126 digital inputs
<b>ECM-LD4D</b>	4 outputs for PWM LED modulation (max. 30V, adjustment range 0-255) 8 digital inputs
<b>ECM-LD9D</b>	9 outputs for PWM LED modulation (max. 30V, adjustment range 0-255) 18 digital inputs
<b>ECM-AO4D</b>	4 analog outputs (0-10V, adjustment range 0-255) 8 digital inputs

CONTROLLER COMMANDS	DEVICE MESSAGES
<b>Messages sent after establishing of the connection</b>	
	DEVICE = DM3 DEVICE = DM6 DEVICE = LD4 DEVICE = LD9 DEVICE = AO4
	LEVEL[<output number>] = <level>
	ON[<output number>] or OFF[<output number>]
	PUSHED[<input number>] or RELEASED[<input number>]
<b>Messages sent after changing of the device state</b>	
	LEVEL[<output number>] = <level>
	ON[<output number>] or OFF[<output number>]
	PUSH[<input number>] or RELEASE[<input number>]
<b>Messages sent as a reply to the controller command</b>	
PING	PING_REPLY
ON[<output number>]	If the channel was OFF: LEVEL[<output number>] = <level> ON[<output number>] If the channel was ON: ON[<output number>]
OFF[<output number>]	If the channel was ON: LEVEL[<output number>] = 0 OFF[<output number>] If the channel was OFF: OFF[<output number>]
INV[<output number>]	If the channel was ON: LEVEL[<output number>] = 0 OFF[<output number>] If the channel was OFF: LEVEL[<output number>] = <level> ON[<output number>]
LEVEL[<output number>] = <level>	LEVEL[<output number>] = <level>
?	For all outputs: LEVEL[<output number>] = <level> ON[<output number>] or OFF[<output number>]
?OUT[<output number>]	LEVEL[<output number>] = <level> ON[<output number>] or OFF[<output number>]

<b>ECM-IO16D</b>	16 universal digital inputs/outputs
<b>EPM-RL6D</b>	6 power relay outputs 6 digital inputs
<b>ECM-RL12LVD</b>	12 signal relay outputs 0 digital inputs

CONTROLLER COMMANDS	DEVICE MESSAGES
<b>Messages sent after establishing of the connection</b>	
	DEVICE = IO16 DEVICE = RL6 DEVICE = RL12
	For all inputs: PUSHED[<input number>] or RELEASED[<input number>]
	For all outputs: ON[<output number>] or OFF[<output number>]
<b>Messages sent after changing of the device state</b>	
	For all inputs: PUSH[<input number>] or RELEASE[<input number>]
	For all outputs: ON[<output number>] or OFF[<output number>]
<b>Messages sent as a reply to the controller command</b>	
PING	PING_REPLY
ON[<output number>]	ON[<output number>] Also for ECM-IO16D: PUSH[<input number>] or RELEASE[<input number>]
OFF[<output number>]	OFF[<output number>] Also for ECM-IO16D: PUSH[<input number>] or RELEASE[<input number>]
PULSE[<output number>]	ON[<output number>] Also for ECM-IO16D: PUSH[<input number>] or RELEASE[<input number>]
PULSE[<output number>]T<time>	ON[<output number>] Also for ECM-IO16D: PUSH[<input number>] or RELEASE[<input number>]
INV[<output number>]	ON[<output number>] or OFF[<output number>] Also for ECM-IO16D: PUSH[<input number>] or RELEASE[<input number>]
?	For all inputs: PUSHED[<input number>] For all outputs: ON[<output number>]
?ALL	For all inputs: PUSHED[<input number>] or RELEASED[<input number>] For all outputs: ON[<output number>] or OFF[<output number>]
?IN	For all inputs: PUSHED[<input number>] or RELEASED[<input number>]
?OUT	For all outputs: ON[<output number>] or OFF[<output number>]
?IN[<input number>]	PUSHED[<input number>] or RELEASED[<input number>]
?OUT[<output number>]	ON[<output number>] or OFF[<output number>]

## EPM-BS3D

6 relay outputs coupled in 3 channels  
6 inputs for direct control

CONTROLLER COMMANDS	DEVICE MESSAGES
<b>Messages sent after establishing of the connection</b>	
	DEVICE = BS3
	For output channels: OPENING[<channel>] or CLOSING[<channel>] or STOPPED[<channel>] For output relays: ON[<relay number>] or OFF[<relay number>]
<b>Messages sent after changing of the device state</b>	
	For output channels: OPENING[<channel>] or CLOSING[<channel>] or STOPPED[<channel>] For relays: ON[<relay number>] or OFF[<relay number>]
<b>Messages sent as a reply to the controller command</b>	
PING	PING_REPLY
ON[<output number>]	ON[<relay number>]
OFF[<output number>]	OFF[<relay number>]
OPEN[<channel>]	OPENING[<channel>]
OPEN[<channel>]T<time>	OPENING[<channel>]
CLOSE[<channel>]	CLOSING[<channel>]
CLOSE[<channel>]T<time>	CLOSING[<channel>]
STOP[<channel>]	STOPPED[<channel>]
?	For all channels: OPENING[<channel>] or CLOSING[<channel>] or STOPPED[<channel>] For all relays: ON[<relay number>] or OFF[<relay number>]
?OUT	For all channels: OPENING[<channel>] or CLOSING[<channel>] or STOPPED[<channel>] For all relays: ON[<relay number>] or OFF[<relay number>]
?OUT[<channel>]	OPENING[<channel>] or CLOSING[<channel>] or STOPPED[<channel>] ON[<relay number of open direction>] or OFF[<relay number of open direction>] ON[<relay number of close direction>] or OFF[<relay number of close direction>]

### ECM-IR4B

4 / 5 IR ports  
255 channels per port

CONTROLLER COMMANDS	DEVICE MESSAGES
<b>Messages sent after establishing of the connection</b>	
	DEVICE = IR4
	For all ports: If the port is configured as transmitting: PORT <port>: ON[<channel>] or PORT <port>: OFF If the port is configured as receiving: PORT <port>: PUSHED[<channel>] or PORT <port>: RELEASED
<b>Messages sent after changing of the device state</b>	
	If the port is configured as transmitting: PORT <port>: ON[<channel>] or PORT <port>: OFF[<channel>] If the port is configured as receiving: PORT <port>: PUSH[<channel>] or PORT <port>: RELEASE[<channel>]
<b>Messages sent as a reply to the controller command</b>	
PING	PING_REPLY
PORT <port>: ON[<channel>]	Stops the transmission (if any) of the IR-command at any channel of the port: PORT <port>: OFF[<transmitting channel>] And starts the transmission of IR-command at defined port and channel PORT <port>: ON[<channel>]
PORT <port>: OFF	Stops any transmission of an IR-command at defined port: PORT <port>: OFF[<transmitting channel>]
PORT <port>: OFF[<channel>]	Stops the transmission of an IR-command at specified port and channel. PORT <port>: OFF[<channel>] If any other channel was transmitting, it will be stopped as well: PORT <port>: OFF[<transmitting channel>]
PORT <port>: PULSE[<channel>]	Stops the transmission (if any) of the IR-command at any channel of the port: PORT <port>: OFF[<transmitting channel>] and transmits the IR-command at specified port and channel during 0.5 sec PORT <port>: ON[<channel>]
PORT <port>: PULSE[<channel>]T<time>	Stops the transmission (if any) of the IR-command at any channel of the port: PORT <port>: OFF[<transmitting channel>] and transmits the IR-command at the port and channel during the time T (in tenths of second) PORT <port>: ON[<channel>]
PORT <port>: IRP-<channel>,<time on>,<time off>	Adds the IR command to the command queue (up to 8). The messages PORT <port>: ON[<channel>] and PORT <port>: OFF[<channel>] will be sent as they are executed in the queue.
PORT <port>: IRCLR	Stops the execution of the queue: PORT <port>: OFF[<channel>].
?	For all ports: If the port is configured as transmitting, defines the transmitting channel: PORT <port>: ON[<channel>] else PORT <port>: OFF If the port is configured as receiving, defines the receiving channel: PORT <port>: PUSHED[<channel>] else PORT <port>: RELEASED
PORT <port>: ?	If the port is configured as transmitting: PORT <port>: ON[<channel>] else PORT <port>: OFF If the port is configured as receiving: PORT <port>: PUSHED[<channel>] else PORT <port>: RELEASED

## ECM-UTM4D

4 universal analog inputs  
3 relay outputs

CONTROLLER COMMANDS	DEVICE MESSAGES
<b>MESSAGES SENT AFTER ESTABLISHING OF THE CONNECTION</b>	
	DEVICE = UTM4
	TI[<channel>] = <temperature value>
	SP[<channel>] = <setpoint value>
	MODE[<channel>] = <mode value> (See description of modes in section "MODE")
	BLOCKING[<channel>] = <blocking value> The "blocking value" can be "EMPTY", "BLOCKED_OFF", "BLOCKED_ON" and corresponds to the blocking state of digital inputs
	ON[<relay number>] or OFF[<relay number>]
	VI[<channel>] = <voltage value> CI[<channel>] = <current value> RI[<channel>] = <resistance value>
<b>Messages sent after changing of the device state</b>	
	TI[<channel>] = <temperature value> BLOCKING[<channel>] = <blocking value> VI[<channel>] = <voltage value> CI[<channel>] = <current value> RI[<channel>] = <resistance value> ON[<relay number>] or OFF[<relay number>]
<b>Messages sent as a reply to the controller command</b>	
PING	PING_REPLY
SP[<channel>] = <setpoint value>	SP[<channel>] = <setpoint value>
MODE[<channel>] = <mode value>	MODE[<channel>] = <mode value> The "mode value" can be: "OFF", "AUTO", "FORCED_OFF", "FORCED_ON" Mode "OFF" has protection against freezing Mode "FORCED_OFF" has no protection against freezing
?	For all channels: TI[<channel>] = <temperature value> SP[<channel>] = <setpoint value>
?ALL	TI[<channel>] = <temperature value> SP[<channel>] = <setpoint value> MODE[<channel>] = <mode value> BLOCKING[<channel>] = <blocking value> VI[<channel>] = <voltage value> CI[<channel>] = <current value> RI[<channel>] = <resistance value> ON[<relay number>] or OFF[<relay number>]
?TI[<channel>]	TI[<channel>] = <temperature value>
?SP[<channel>]	SP[<channel>] = <setpoint value>
?MODE[<channel>]	MODE[<channel>] = <mode value>
?BLOCKING[<channel>]	BLOCKING[<channel>] = <blocking value>
?OUT[<channel>]	ON[<relay number>] or OFF[<relay number>]
?IN[<channel>]	VI[<channel>] = <voltage value> CI[<channel>] = <current value> RI[<channel>] = <resistance value>

<b>ECM-DTS16D</b>	8 channels in single-ended connection 16 channels in bus connection
-------------------	--

CONTROLLER COMMANDS	DEVICE MESSAGES
<b>Messages sent after establishing of the connection</b>	
	DEVICE = DTS16
	TI[<channel>] = <temperature value>
<b>Messages sent after changing of the device state</b>	
	TI[<channel>] = <temperature value>
<b>Messages sent as a reply to the controller command</b>	
PING	PING_REPLY
?	For all channels: TI[<channel>] = <temperature value>
?TI[<channel>]	TI[<channel>] = <temperature value>