

# ipEther232.IO

## Command Reference

### Version 12.1

| Function                         | UDP Port |
|----------------------------------|----------|
| Control Port                     | 3497     |
| Data Port (Digital Input/Output) | 3498     |
| RS232 Receive/Transmit           | 3499     |

#### General Commands (Port 3497)

| Command                   | Response        | Description  | Value Range | Default |
|---------------------------|-----------------|--|-------------|---------|
| “Open”                    | “ok” or “error” | Open the connection or keep it alive.                              |             |         |
| “OpenEx”                  | “ok” or “error” | Open the connection exclusively or keep it alive.                  |             |         |
| “Open <i>password</i> ”   | “ok” or “error” | Like “Open” but with password.                                     |             |         |
| “OpenEx <i>password</i> ” | “ok” or “error” | Like “OpenEx” but with password.                                   |             |         |
| “Close”                   | “closed”        | Close the connection.  |             |         |
| “?” or “Help”             | ....            | Return a list of all recognised commands and the current settings. |             |         |

[ ... ] optional arguments

Up to four clients are able to open a connection to the device at the same time, unless the first client opens an exclusive connection – in this case no further clients are able to connect to the device.

Only clients that have an open connection can use any of the non-general commands described on the following pages. Non-general commands from senders that are not recognized as a client will be ignored.

**IO Commands (Port 3497)**

| Command                       | Answer   | Description   | Value Range                                | Default |
|-------------------------------|--|---|--|---------|
| <b>“SampleRate [value]”</b>   | <b>“SampleRate value”</b>  | Set the sample rate in milliseconds.  | 1...255                                    | 10      |
| <b>“TransmitRate [value]”</b> | <b>“TransmitRate value”</b>  | Set the transmit rate in milliseconds.<br>0 = no cyclic transmission                    | 0 = off<br>50..65535                       | 2000    |
| <b>“Debounce [value]”</b>     | <b>“Debounce value”</b>  | Number of consistent samples before the pin state is accepted.                          | 1..15                                      | 3       |
| <b>“OnChange [value]”</b>     | <b>“OnChange value”</b>  | Transmit if there are any changes, but at most every 100 ms.                            | 0 = off<br>1 = on                          | 1       |
| <b>“RC counter [value]”</b>   | <b>“RC counter value”</b>  | Decreases the “counter” by “value”. Without the value parameter it will be set to zero. | Counter 0..5<br>Value<br>0..<br>4294967295 | 0       |
| <b>“Counter [value] ”</b>     | <b>“Counter value”</b>   | Transmit counters [on off].   | 0 = off<br>1 = on                          | 1       |
| <b>“Events [value] ”</b>      | <b>“Events value”</b>  | Transmit events [on off].   | 0 = off<br>1 = on                          | 1       |
| <b>“HasOutput [value]”</b>    | <b>“HasOutput value”</b>   | Tells the device if it is equipped with an output module or not.                        | 0 = no<br>1 = yes                          | 0       |
| <b>“Output pin value ”</b>    | <b>“Output pin value”</b><br>or <b>“error”</b> if<br>HasOutput is 0. | Set the output “pin” (0 or 1) on or off.  | 0 = off<br>1 = on                          |         |
| <b>“OutPulse pin ms ”</b>     | <b>“OutPulse pin ms ”</b><br>or <b>“error”</b> if<br>HasOutput is 0. | Pulses the output (0-1-0) on “pin” (0 or 1) for the given duration in milliseconds.     | 10..65535<br>ms                            |         |

[ ... ] optional arguments

## RS232 Commands (Port 3497)

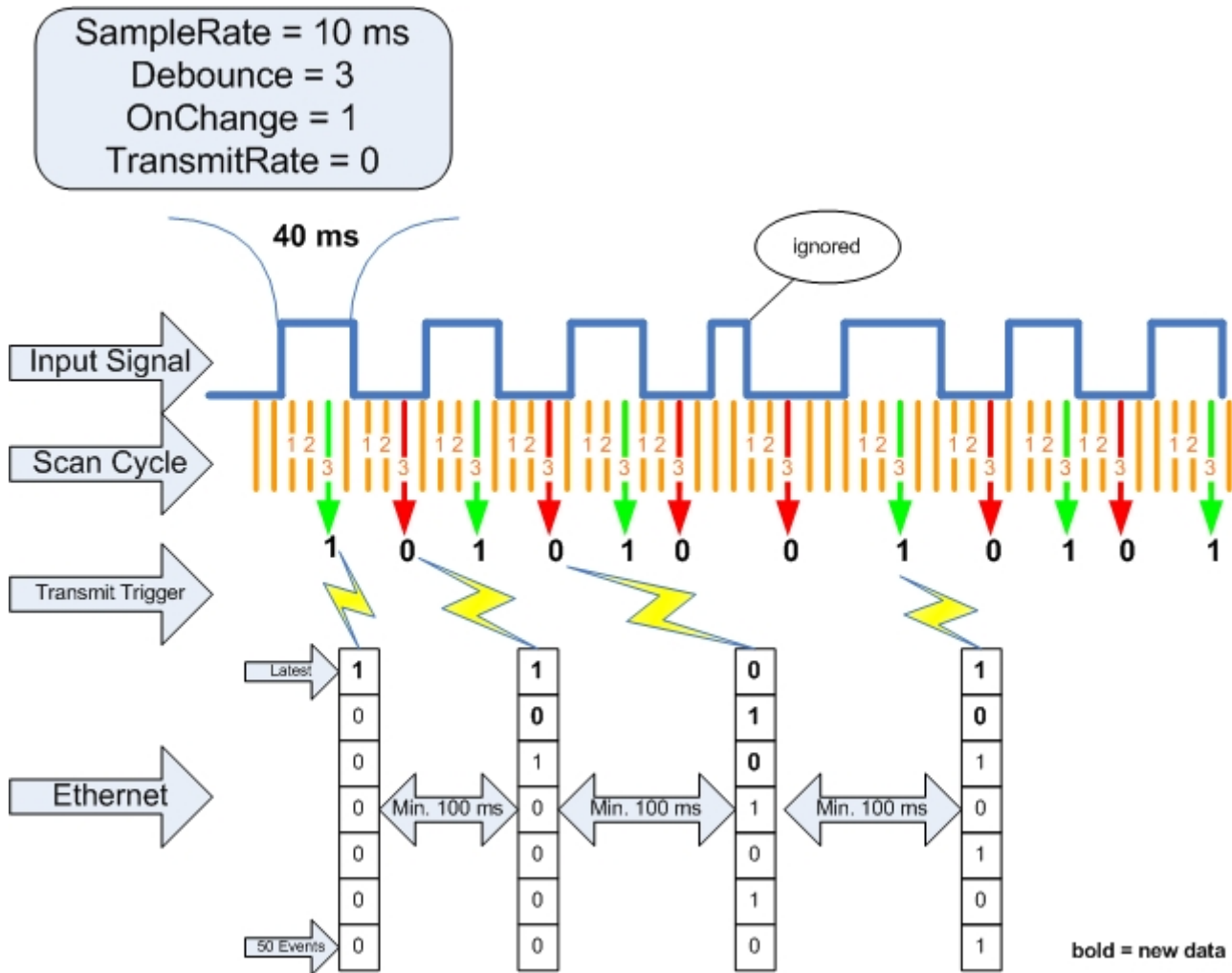
| Command                                     | Answer                                      | Description   | Value Range   | Default    |
|---|---|---|---|------------|
| <b>“Mode</b> <i>baud,data,parity,stop</i> ” | <b>“Mode</b> <i>baud,data,parity,stop</i> ” | Set UART parameters.  | Baud = 2400..115200<br>Data = 7   8<br>Parity = n   e   o   1   0<br>Stop = 1   2 | 9600,8,n,1 |
| <b>“Mode”</b>                               | <b>“Mode</b> <i>baud,data,parity,stop</i> ” | Get the current settings.   |   |            |
| <b>“FAF</b> <i>[value]</i> ”                | <b>“FAF</b> <i>value</i> ”                  | Fire-And-Forget.<br>If this mode is off, the client application has to acknowledge that it has received the data. | 0 = off<br>1 = on   | 1          |
| <b>“RXRate</b> <i>[value]</i> ”             | <b>“RXRate</b> <i>value</i> ”               | Number of milliseconds before transmitting the received data.   | 50..65535 ms  | 300        |
| <b>“CharTimeout</b> <i>[value]</i> ”        | <b>“CharTimeout</b> <i>value</i> ”          | Number of milliseconds with no received bytes before the data is transmitted via Ethernet (gap detection).        | 0 = off<br>5..255 ms  | 0          |

[ ... ] optional arguments

## UDP Ports Used for Responses and Data Messages

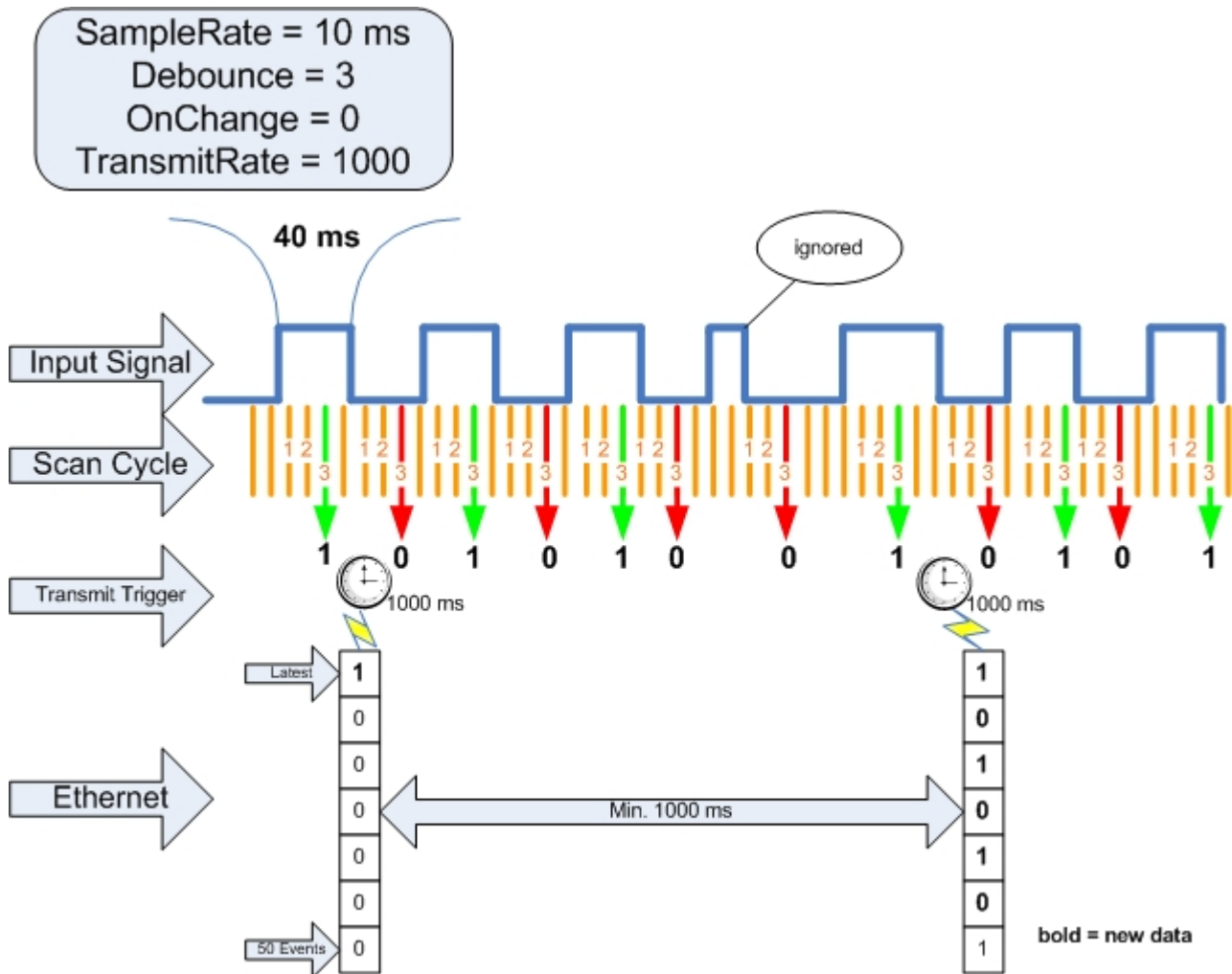
- UDP datagrams from port 3497 are responses to command messages.
- UDP datagrams from port 3498 are I/O data messages.
- UDP datagrams from port 3499 are RS232 data messages.

## Input Pin Scan with Spontaneous Transmission



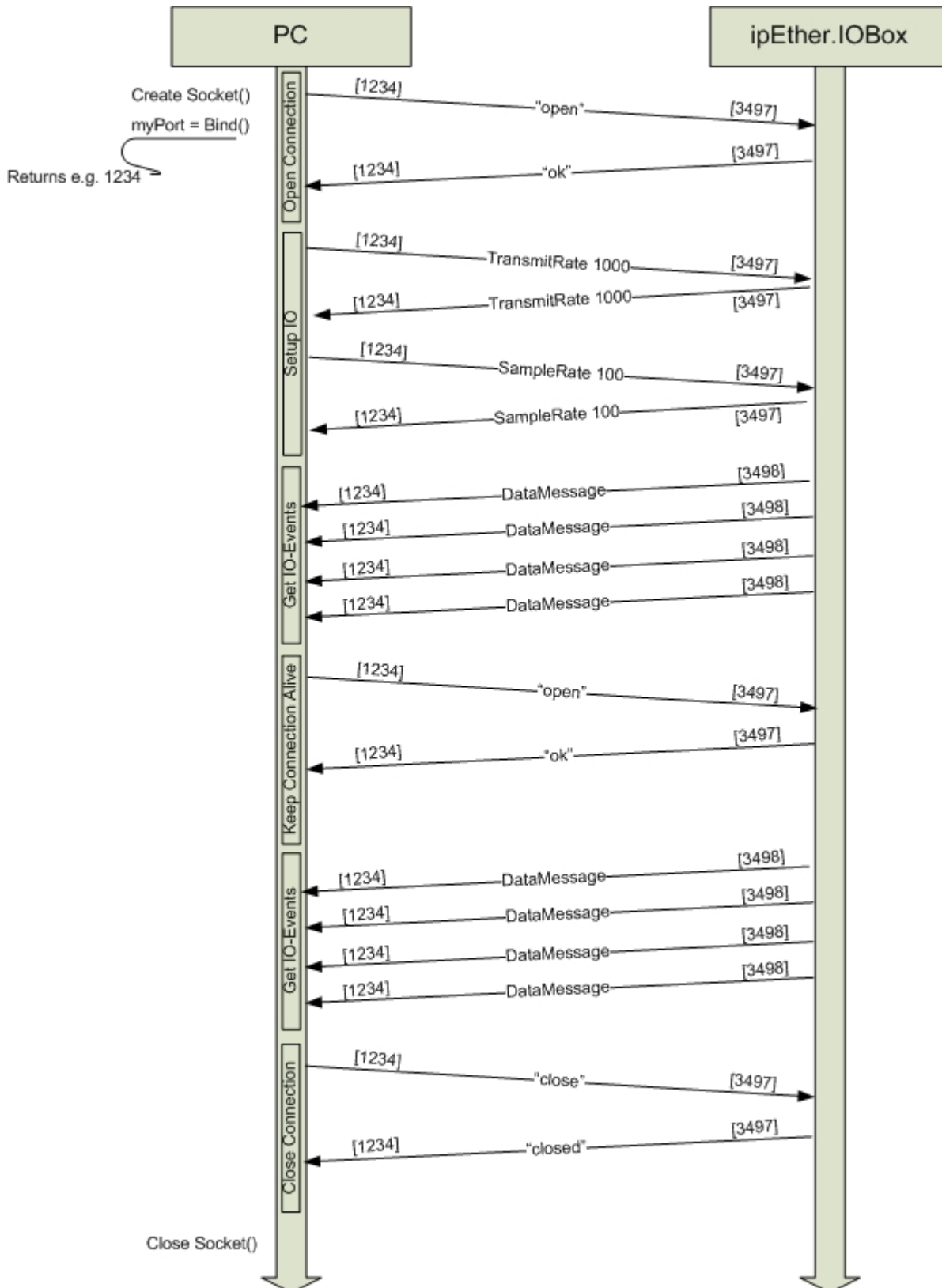
With this configuration you will get more network traffic but a prompt notification about pin changes.

## Input Pin Scan with Cyclic Transmission



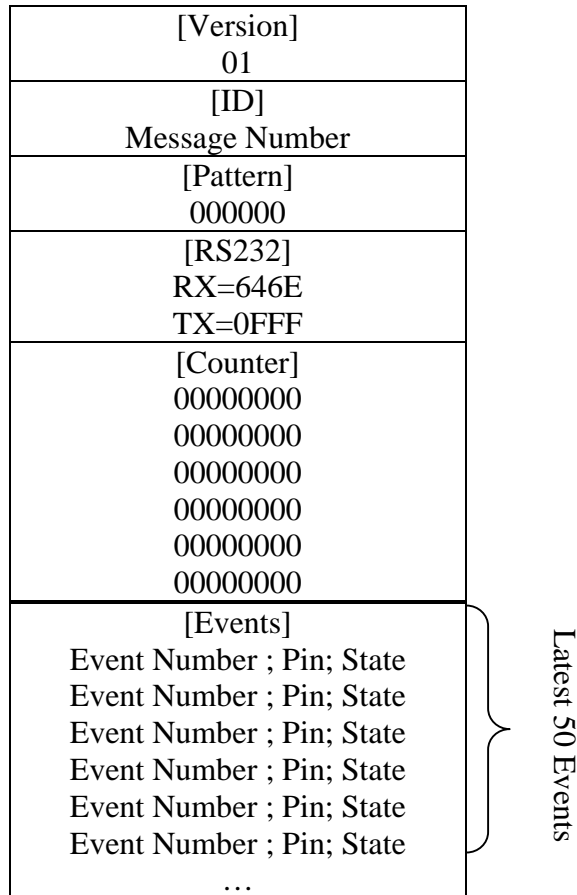
With this configuration you will get less network traffic but events can be “TransmitRate” milliseconds old.

## Communication Flow



## Data Message Format I/O (Port 3498)

- The UDP data message contains ASCII encoded lines.
- The line separator is '\n' (0x0A).
- The message is separated into sections: *[SectionName]*
- This document describes structure version 1.



|                       |  |
|-----------------------|--|
| <b>Version</b>        | Version number of this structure layout.   |
| <b>Message Number</b> | A 4 digit hexadecimal number. Each UDP message increments this number. It runs from 0 to 0xFFFF and then starts at 0 again.<br>With this number lost messages can be detected.                         |
| <b>Pattern</b>        | The current input/output pattern (binary 6 bits) from pin 0 to pin 5.  |
| <b>RX</b>             | Free bytes in receive buffer (hexadecimal).  |
| <b>TX</b>             | Free bytes in transmit buffer (hexadecimal).   |
| <b>Counter</b>        | Internal counter for each pin. A counter is increased every time an event changes the corresponding pin to high (1).   |
| <b>Event Number</b>   | Each event (a relevant pin change) gets an increased event number.<br>This number initially starts with 1 and eventually wraps around from 0xFF to 0.<br>With this number lost events can be detected. |
| <b>Pin</b>            | The number of the pin that caused this event. Pin numbers range from 0 to 5.   |
| <b>State</b>          | The state of the pin. 0 = off / 1 = on   |

## RS232 Message Format (Port 3499)

If the fire-and-forget mode is active (FAF 1), received RS232 data will be transferred “raw”, ie. the pure data received via RS232 will be sent to clients from UDP port 3499.

But if the fire-and-forget mode is not active (FAF 0), received RS232 data will be embedded in a structure with the following format:

- The line separator is ‘\n’ (0x0A).
- The message is separated into two sections: [RS232] and [Data]

|  |
|--|
| [RS232]<br>Pos=0x0009<br>Len=0x0005<br>Overrun=0 |
| [Data]<br>Raw Data                               |

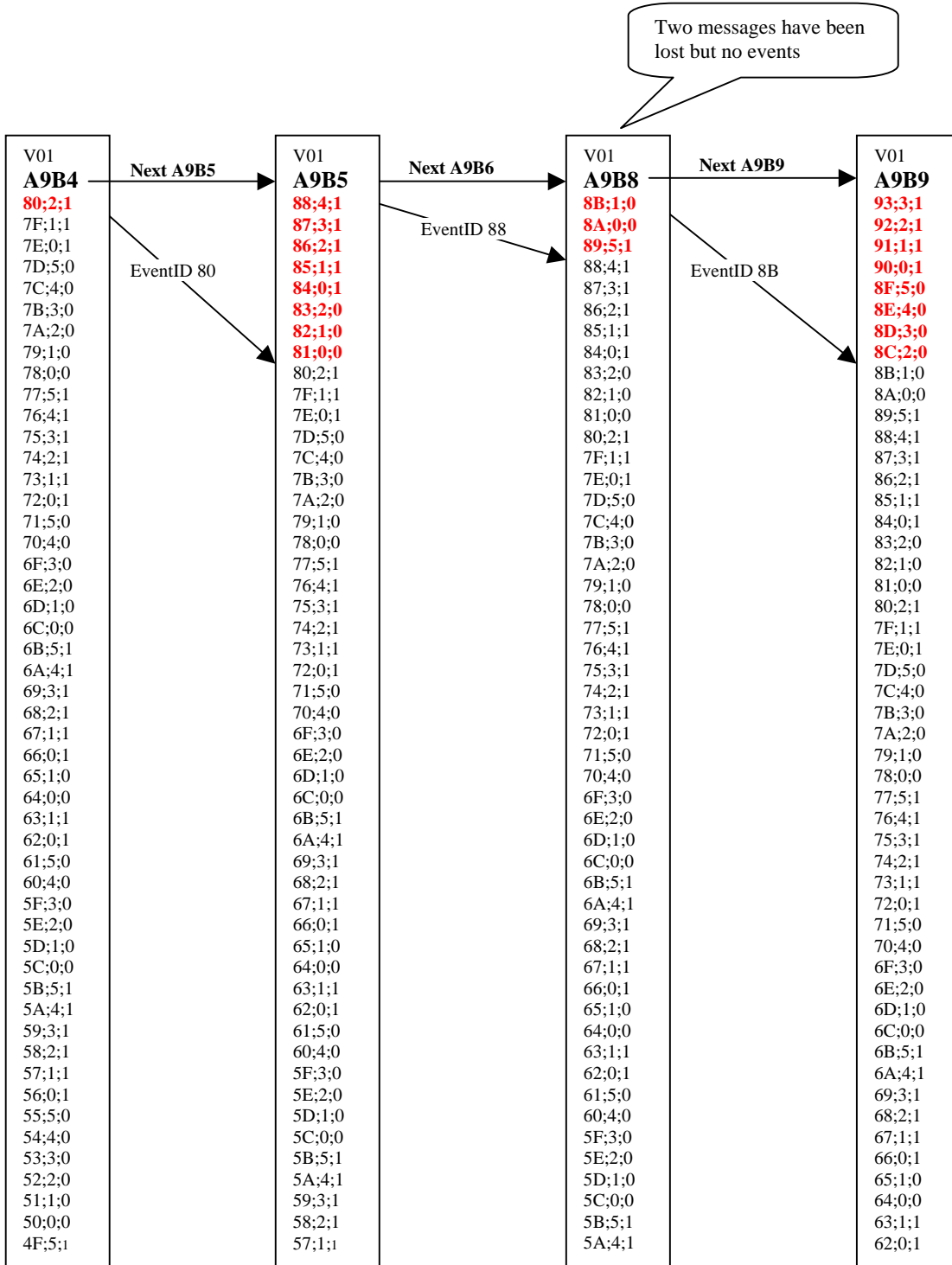
|                 |   |
|-----------------|---|
| <b>Pos</b>      | Position in the internal ring buffer.                           |
| <b>Len</b>      | The amount of data bytes being transferred.                     |
| <b>Overrun</b>  | If 1 it indicates that a buffer overrun occurred in the device. |
| <b>Raw Data</b> | The raw data that has been received from RS232.                 |

When the fire-and-forget mode is not used, clients have to acknowledge that they received the RS232 data message. The acknowledgment message has to be sent back to UDP port 3499 and looks like this:

|   |
|---|
| [RS232]<br>Pos=0x0009<br>Len=0x0005<br>[Data] |
|---|

|              |   |
|--------------|---|
| <b>RS232</b> | The RS232 section contains the Pos and Len lines but Overrun is not used in the acknowledgment. |
| <b>Pos</b>   | Position relative to the 0x7000 byte sized ring buffer.   |
| <b>Len</b>   | The amount of data bytes received.  |
| <b>Data</b>  | Apart from a newline (‘\n’) the Data section is empty.  |

## Sample Data Message



red = new event