Command interface

The command interface can be used for accessing Netzer with client programs.

Functional principle

The command interface is message based, i.e. defines the format of the exchanged messages between Netzer and a client program. The communication channel is mostly independent from the message. Hence, multiple channels are implemented from where the command interface can be used. Which channels are implemented depends on the Netzer firmware version. Possible channels are the command server, CGI and WebSocket.

One message consists of a command and a value.

Messages without a value can be used for reading parameters.

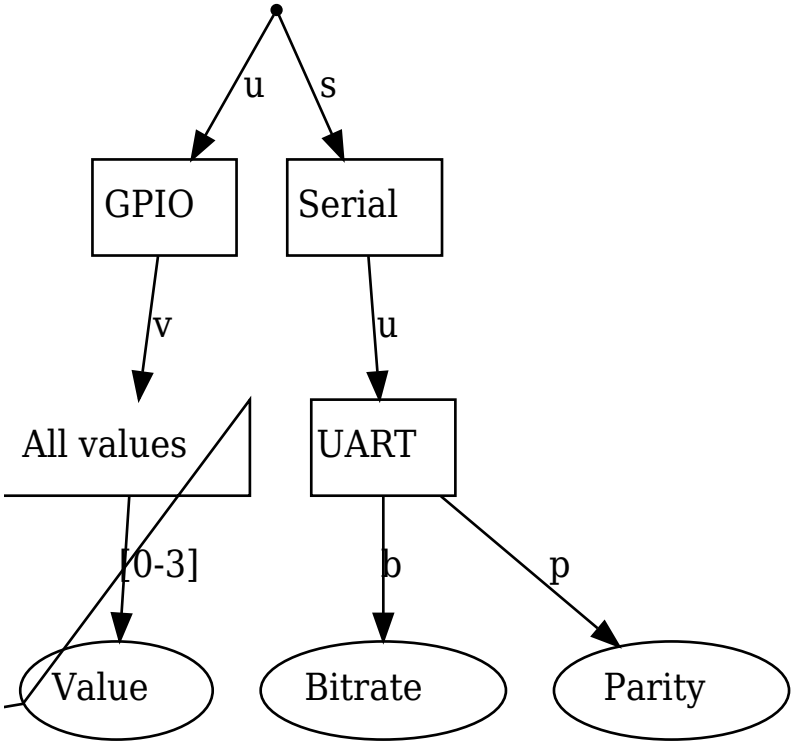
Client to Netzer	Netzer to client
`sup`	`{"s":{"u":{"p":"0"}}}`

Messages with value to write.

Client to Netzer	Netzer to client
`sup=1`	`{"s":{"u":{"p":"1"}}}`

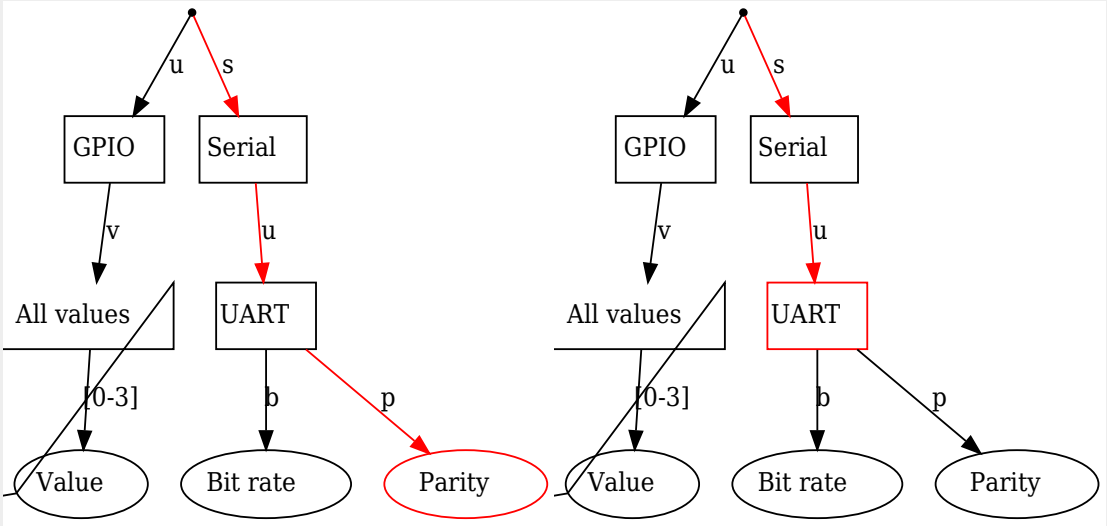
The command tree

The command tree is the base for all commands. A command is a (not empty) sequence of alphanumeric characters (lower case letters and numbers) which describes a path within the command tree.



Write commands (messages with value) have to end on a tree leaf (i.e. identify exactly one parameter), because only one parameter can be written at once.

Read commands also can also end at an inner node of the command tree. Hence, all nodes beneath the read node can be read at once.



Read:

Client to Netzer	Netzer to client
`sup`	`{"s":{"u":{ "p":"0"}}}`

Read:

Client to Netzer	Netzer to client
`su`	`{"s":{"u":{ "b":"5","p" :"0"}}}`

Write:

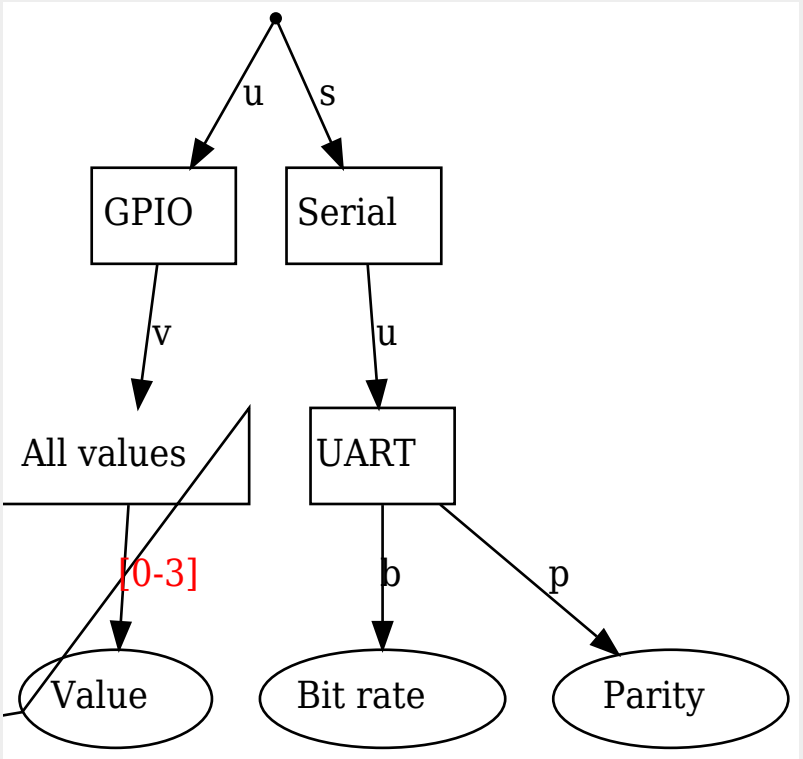
Client to Netzer	Netzer to client
sup=1	{"s":{"u":{ "p":"1"}}}

Write:

Not supported

Edges with numerical values

Numeric edges are used as represenation for continuous numeric intervals. Hence, adding each single number as a command tree node is prevented. Such edges can be recognized by square brackets which stand for the possible hexadecimal interval for this edge.



For the edge with *[0-3]* the hexadecimal numbers `0`, 1, 2 and 3 are possible.

Multiple numerical number edges within a command are interpreted as continuous interval. The interval limits are build through the concatenation of the numbers within the square brackets.

The path leads via two edges with *[0-1]* and *[0-c]*. This means the matematical interval `[0x00, 0x1c]`.

The following commands are possible: `00`, `01`, `02`, `03`, `04`, `05`, `06`, `07`, `08`, `09`, `0a`, `0b`, `0c`, `0d`, `0e`, `0f`, `10`, `11`, `12`, `13`, `14`, `15`, `16`, `17`, `18`, `19`, `1a`, `1b` and `1c`.

Channel specific commands

Beyond the command tree, which depends on the Netzwer firmware, also channel specific commands may exist. Those commands always start with `z`.

Messages from client to Netzer

Messages from client to Netzer consist of a string containing the command and an optional value like `command=value`. Per message only one parameter can be written.

The exact format syntax for connecting command and value depends on the used channel. But normally it is the format `command=value`.

Example WebSocket channel

Client to Netzer Netzer to client

```
sup=1          {"s":{"u":{"
               p":"1"}}}
```

Messages from Netzer to client

Messages from Netzer to client use the JSON format. Hence, it is easy to use a JSON parser within a browser to convert the message into a Javascript object.

The message is a JSON object type and contains one or more properties. Each property represents an edge within the command tree. The property key is the edge character. Inner node edge values are also objects with one or more properties. Leaf node edge values are simple data types like boolean or strings. The addressed parameter specifies the type of those leaf values. Numeric parameters are normally formated as strings.

From:
http://www.mobacon.de/dokuwiki/ - **MoBaCon**

Permanent link:
http://www.mobacon.de/dokuwiki/doku.php?id=en:netzer:commandinterface

Last update: **2025/06/11 20:42**