Kommandoschnittstelle

Um mit Clientprogrammen auf den Netzer zuzugreifen, stellt dieser die Kommandoschnittstelle zur Verfügung.

Funktionsprinzip

Die Kommandoschnittstelle ist nachrichtenbasiert. D.h. sie definiert das Format von Nachrichten, die zwischen Netzer und Clientprogramm ausgetauscht werden. Der Kommunikationskanal ist davon weitestgehend unabhängig. Daher gibt es auch mehrere Kanäle, über die die Kommandoschnittstelle genutzt werden kann. Die zur Verfügung stehenden Kanäle hängen von der Version des Netzers ab. Mögliche Kanäle sind der Kommandoserver, CGI und WebSocket.

Eine Nachricht besteht aus einem Befehl und optional einem Wert.

Nachrichten ohne Wert dienen zum Auslesen von Parametern.

Client an Netzer

```
sup
```

Netzer an Client

```
{"s":{"u":{"p":"0"}}}
```

Nachrichten mit Wert zum Schreiben.

Client an Netzer

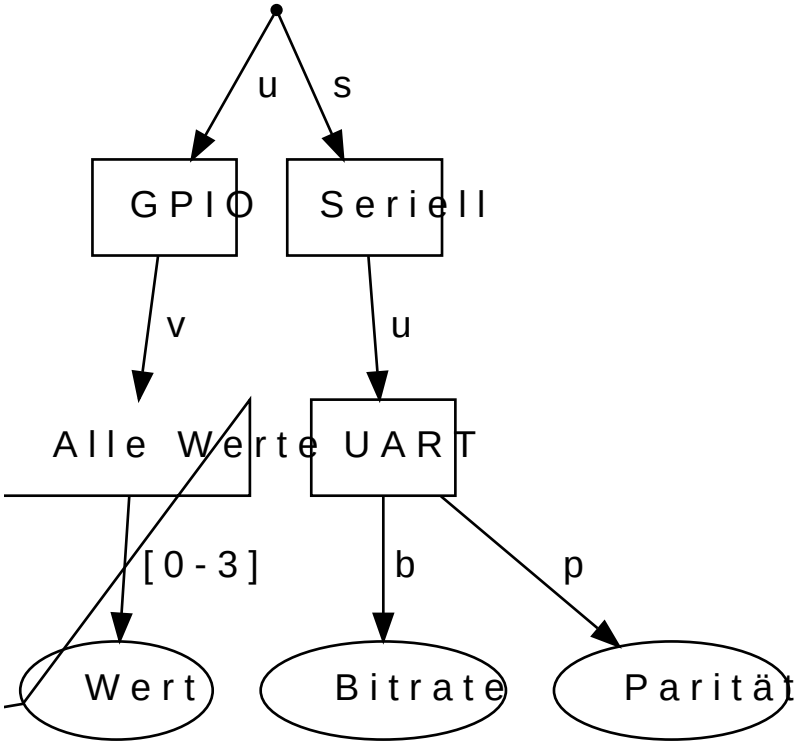
```
sup=1
```

Netzer an Client

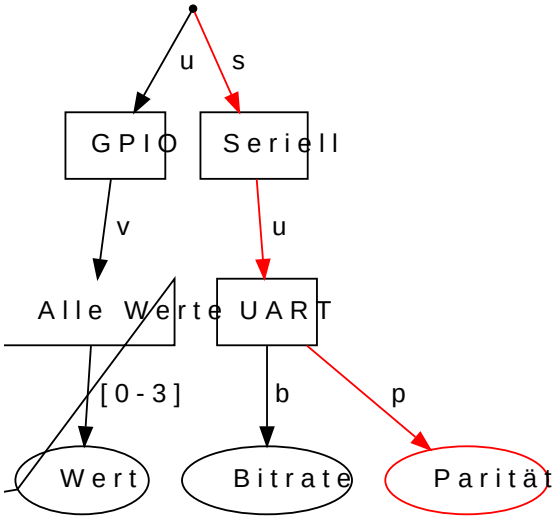
```
{"s":{"u":{"p":"1"}}}
```

Der Befehlsbaum

Die Grundlage der Befehle ist der Befehlsbaum. Ein Befehl ist eine (nicht leere) Folge von alphanumerischen Zeichen (Kleinbuchstaben und Zahlen), die einen Pfad im Befehlsbaum beschreibt.



Schreibbefehle (d.h. Nachrichten mit Wert) müssen in einem Blatt des Befehlsbaums enden (d.h. genau einen Parameter identifizieren), da nur einzelne Werte geschrieben werden können. Lesebefehle hingegen können auch in einem inneren Knoten des Befehlsbaums enden. Dadurch werden alle Parameter des Teilbaums unterhalb des Knotens gleichzeitig ausgelesen.



Lesen:

Client an Netzer

```
sup
```

Netzer an Client

```
{"s":{"u":{"
p":"0"}}}
```

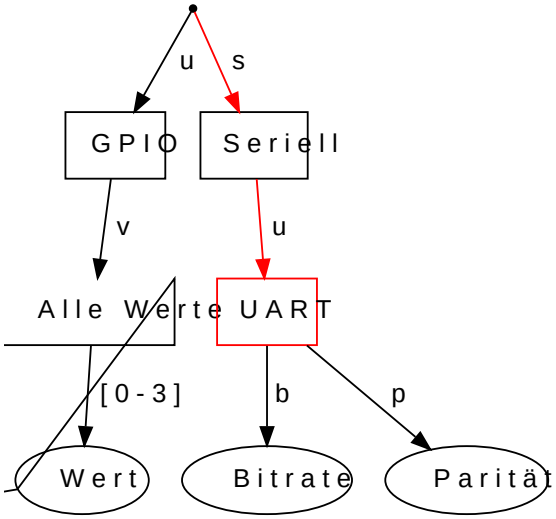
Schreiben:

Client an Netzer

```
sup=1
```

Netzer an Client

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



Lesen:

Client an Netzer

```
su
```

Netzer an Client

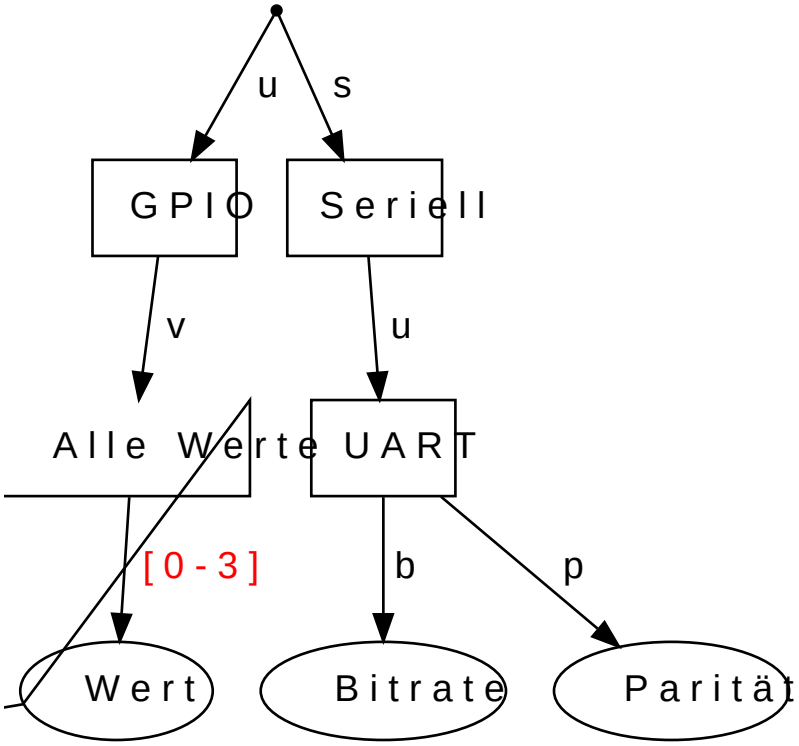
```
{"s":{"u":{"
b":"5","p":"
0"}}}
```

Schreiben:

nicht möglich

Kanten mit numerischen Werten

Um zu vermeiden, dass für zusammenhängende numerische Intervalle jede Ziffer einzeln in den Befehlsbaum aufgenommen werden muss, gibt es Kanten mit numerischen Werten. Diese erkennt man an den eckigen Klammern, die für das geschlossene Intervall von hexadezimalen Ziffern stehen, die für diese Kante möglich sind.



Für die Kante mit *[0-3]* sind die Hexadezimalziffern `0`, `1`, `2` und `3` möglich.

Mehrere Kanten mit numerischen Werten innerhalb eines Befehls werden als kontinuierliches Intervall interpretiert, nicht als einfache Konkatenation der Ziffern. Nur die Intervallgrenzen werden durch die Konkatenation der Ziffern innerhalb der eckigen Klammern gebildet.

Führt ein Pfad über zwei Kanten mit *[0-1]* bzw. *[0-c]*, so entspricht dies dem mathematischen Intervall `[0x00, 0x1c]` und steht für die möglichen Befehlsteile `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` und `1c`.

Kanalspezifische Befehle

Neben den Befehlen aus dem Befehlsbaum, die nur von der Netzerversion, aber nicht vom

verwendeten Kanal abhängen, gibt es auch kanalspezifische Befehle. Sie beginnen stets mit `z`.

Nachrichten vom Client zum Netzer

Nachrichten, die vom Client zum Netzer gesendet werden, bestehen aus einem String, der den Befehl enthält und optional einem Wert (z.B. in der Form `Befehl=Wert`). Pro Nachricht kann nur ein Parameter geschrieben werden.

Wie genau Befehl und Wert miteinnader verbunden werden, hängt von dem benutzen Kanal ab. In der Regel werden sie als `Befehl=Wert` geschrieben.

Beispiel WebSocket

Client an Netzer

```
sup=1
```

Netzer an Client

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

Nachrichten vom Netzer zum Client

Nachrichten, die vom Netzer zum Client gesendet werden, folgen dem JSON-Format. So kann zum Beispiel ein Browser die Nachrichten mit Hilfe eines JSON-Parsers direkt in ein JavaScript-Objekt überführen.

Die Nachricht selbst ist vom JSON-Datentyp Objekt und enthält eine oder mehrere Eigenschaften. Jede Eigenschaft entspricht einer Kante im Befehlsbaum. Dabei besteht der Schlüssel der Eigenschaft aus dem Zeichen, das der Kante entspricht. Führt die Kante zu einem inneren Knoten, so ist der Wert wiederum ein Objekt mit einer oder mehreren Eigenschaften. Führt die Kante zu einem Blatt, so ist der Wert ein anderer JSON-Datentyp als Objekt. Welcher Datentyp genau, hängt von dem Parameter ab, den das Blatt identifiziert. Üblich sind Zeichenketten und boolsche Werte. Auch Parameter die an sich numerische Werte besitzen, werden in der Regel als Zeichenketten dargestellt.