Simple Railroad Command Protocol 0.8.2

Torsten Vogt, Martin Ostermann, Kurt Harders, Olaf Schlachter, Matthias Trute (Maintainer), Tobias Schlottke, Edbert van Eimeren, Stefan Bormann, Michael Reukauff, Martin Wolf, Matthias Peick, Martin Schönbeck 2000, 2004 (05.6.2004)

Das SRCP ist ein TCP basiertes Internetprotokoll zur Steuerung und Programmierung von (digitalen) Modelleisenbahnanlagen.

Inhaltsverzeichnis

1 Einführung

Der Einsatz digitaler Modellbahnausrüstung ist bislang sehr stark von der eingesetzten Hardware geprägt. Es gibt eine Übereinkunft zwischen einigen Herstellern, auf der Ebene der Bauelemente gegenseitige Einsetzbarkeit zu erreichen. Diese Übereinkunft ist jedoch nicht allgemein akzeptiert. Eine gemeinsame Steuersprache existiert nicht, jedes System benutzt seine eigene.

Um dem Modellbahner leistungsfähige Programme bereitstellen zu können, muß die Software deshalb für alle derzeit eingesetzten und zukünftigen Digitalsysteme extra angepaßt und getestet werden.

Darüberhinaus ist ein Multiuserbetrieb nur eingeschränkt möglich, wenn überhaupt vorhanden.

Mit dem Simple Railroad Command Protocol werden folgende gleichrangigen Ziele verfolgt:

- Berücksichtigung verschiedener Digitalsysteme mit einer einheitlichen Steuersprache im Multiuserbetrieb im Netzwerk.
- Weite Skalierfähigkeit im Bereich von spontan aufgebauter Teppichbahn", stationärer Großanlage und Einsatz bei Modellbahntreffen, bei denen individuell gestaltete Module zusammengefügt werden.
- Beachten von unterschiedlichen Sicherungsstrategien; von "keine"bis meine eigene".
- Software muß auch vom ïnteressierten Laienërstellbar sein.
- Ein SRCP Server abstrahiert die Ansteuerung der Anlage. Alle Informationen werden entweder aufgrund von direkten Abfragen der Anlage, wo immer möglich, bzw. anhand der Kommandohistorie bereit gestellt.

Vorliegender Text definiert sowohl das Protokoll an sich als auch Syntax und Semantik der zu verwendenden Geräte.

1.1 Begriffliches

In diesem Text werden die Konventionen entsprechend RFC 1122 in einer deutschen Übersetzung verwendet. Im einzelnen sind dies MUSS", SSOLL", ËMPFOHLEN", "KANNünd ÖPTIONAL".

MUSS"(bzw. ËRFORDERLICH"bzw. die Verneinung "DARF NICHT") kennzeichnet eine zwingend einzuhaltende Vorschrift. SSOLLünd ËMPFOHLENßtehen für eine pragmatische Empfehlung, die auch ignoriert werden darf, aber erst nach gründlicher Erwägung. "KANNünd ÖPTIONALßtehen für Eigenschaften, die man guten Gewissens weglassen kann.

2 Überblick

2.1 Das Große Bild

SRCP vermittelt eine Client-Serverkommunikation zur Steuerung einer Modellbahn sowie ein einheitliches Datenmodell für die aktiven Elemente der Modellbahn.

Der Server ist diejenige Komponente, die mit der Anlage verbunden ist. Ein Client nimmt mit dem Server Kontakt auf, um Befehle an die Anlage zu senden und Informationen von dort zu erhalten. Der Client kann auf dem gleichen Rechner wie der Server laufen, er kann aber auch über das Internet vom anderen Ende der Welt aus die Anlage steuern. Für den Client ist das verwendete Modellbahnsystem von untergeordneter Bedeutung. Es ist Aufgabe des Servers, die Befehle des SRCP geeignet umzusetzen.

Ein SRCP Server hat kein Wissen über die konkrete Anlage. Er kennt weder die Topologie, noch ist er (im allgemeinen) darüber informiert, welche Ausrüstung installiert ist. Er übermittelt die Befehle äuf Verdacht". Seine Aufgabe ist es

jedoch, alle verfügbaren Informationen so exakt wie nur irgend möglich zu beschaffen. Hierbei bedient er sich seines Wissens über die Möglichkeiten und Grenzen der von ihm angesteuerten Anlage. Seine Angaben sind immer als "bestes Wissen"über den Anlagenzustand anzusehen. Wann immer die Anlage die Daten liefern kann, wird der Server dies nutzen.

SRCP erfüllt keine formalen Echtzeitforderungen zur Steuerung. Inbesondere gibt es keine garantierten Aktionszeiten.

2.2 SRCP Server und Central Units

Eine Central Unit (CU) ist in der klassischen Modellbahnsteuerung dasjenige Element, das die alleinige Steuerhoheit über die Anlage hat. Jede CU verfügt über Möglichkeiten, die unterschiedlichen aktiven Elemente zu adressieren und an sie Informationen zu übermitteln bzw. von ihnen Informationen abzufragen.

Ein SRCP Server SOLL genau eine Central Unit im herkömmlichen Sinn repräsentieren. Wenn ein SRCP-Server mehrere CU verwalten oder bereitstellen kann, so ist dies zulässig. Eine Trennung zwischen den CU erfolgt dann durch die Busnummer. Wenn an einem Rechner zwar mehrere CU angeschlossen sind, es jedoch keinen gemeinsamen SRCP Server gibt, so sind mehrere (ggf. unterschiedliche) SRCP-Server auf unterschiedlichen TCP Ports (s.u.) zu betreiben. Genau einer dieser Server MUSS den Standardport bereitstellen.

2.3 Netzwerkverbindung

Das SRCP basiert auf den abgesicherten Datenübertragungstechniken des TCP. Hierbei wird ein Port belegt. Derzeit SOLL der Port 12345 benutzt werden. Eine Registrierung bei der IANA kann dies zukünftig verlagern und dem Port auch einen offiziellen Namen verleihen. Der vorläufige Name ist `srcp 12345`tcp/.

2.4 Lexikalische Einheiten

Alle Zeichencodes werden in ihrer dezimalen Form als # gefolgt vom numerischen Wert des Zeichens beschrieben.

Die gesamte Kommunikation besteht aus den Zeichen des 7Bit ASCII Zeichensatzes im Bereich #32 bis #127 (einschl. der Grenzen). Zusätzlich sind die Zeichen #9 (TAB) für Leerraum, #10 und #13 (CR, LF) für das Zeilenende zulässig. Alle weiteren Zeichen (insb. Zeichen mit einem Codewert $>$127) werden in eingehenden Daten ignoriert und entfernt. In ausgehenden Daten können diese als Bestandteil von Daten enthalten sein, im SRCP werden sie nicht benutzt. Die Zeichen #9 und #32 werden als Whitespace angesehen und gelten auch bei mehrfacher unmittelbarer Wiederholung (auch gemischt) als ein Whitespace.

Zahlen werden mindestens als vorzeichenbehaftete 32Bit Ganzzahlen verarbeitet. Dieser Wertebereich kann in Sonderfällen auch erweitert werden. Führende Nullen sind nicht signifikant. Gleitkommazahlen werden NICHT verwendet.

2.5 Kommandos

Kommandos werden im Kommandomodus und im Handshake vom Client an den Server gesendet. Der SRCP Server bearbeitet das Kommando und generiert eine Antwort, die an den Client gesendet wird.

Kommandos bestehen aus einem Kommandowort, gefolgt von einer durch Whitespace abgetrennten Kommandoparameterliste. Die Elemente der Kommandoliste werden ihrerseits durch Whitespace voneinander getrennt. Das Ende eines Kommandos ist das Zeilenende. Die Fortsetzung eines Kommandos in der Folgezeile oder mehrere Kommandos in einer Zeile sind nicht zulässig. Elemente der Parameterliste, die Whitespace enthalten, sind nicht zulässig.

Enthält eine Parameterliste mehr Elemente als für den betreffenden Befehl spezifiziert ist, so MÜSSEN die überzähligen Elemente ignoriert, die Befehlsliste verkürzt und dann bearbeitet werden. Enthält die Parameterliste zuwenig

Elemente, so ist eine Fehlermeldung zu generieren. Sind für einen Befehl mehrere, verschieden lange Parameterlisten definiert, ist immer die längstmögliche Listeninterpretation anzuwenden.

Das Zeilenende besteht aus dem Zeichen #10 (\n, LF). Ein vorangestelltes #13 (\r, CR) ist zulässig. Das Zeichen #13 wird ignoriert. Eine einzelne Zeile DARF NICHT inkl. des Zeilendes eine maximale Länge von **1000** Zeichen überschreiten. Vom Server gesendete Informationen unterliegen den gleichen Bedingungen. Das Zeichen #13 KANN unmittelbar vor dem #10 gesendet werden.

Bei allen Worten wird zwischen Groß- und Kleinschreibung unterschieden. Die Kommandos und Antworten des SRCP sind immer groß geschrieben. Einzige Ausnahme sind erläuternde Texte bei den Fehlermeldungen und Informationen, die von den Geräten übermittelt werden.

2.6 Antworten

Ein Server MUSS auf jedes Kommando eines Clients mit einer Antwort reagieren. Diese wird auf der gleichen TCP Verbindung an den Client gesendet. Eine Antwort SOLL einzeilig sein. Durch das Zeichen \ (Backslash, #92) unmittelbar vor dem LF (resp. dem CRLF) wird signalisiert, das die Antwort eine weitere Zeile umfaßt. In diesem Fall ist die Zeichenfolge \LF (resp. \CRLF) dem Whitespace gleichzusetzen. Eine Antwort kann so auf beliebig viele Zeilen verteilt werden. Eine einzelne Zeile DARF inkl. des Zeilenendes eine maximale Länge von **1000** Zeichen NICHT überschreiten.

Eine Antwort ist entweder eine Eingangsbestätigung eines Kommandos (Quittung), eine Fehlermeldung oder das Ergebnis des Kommandos. Wenn die Antwort eine Fehlermeldung ist, DARF das Kommando NICHT ausgeführt werden.

Eine Antwort ist mit dem wirksamen Zeilenende abgeschlossen. Das Übermitteln von mehr als einer Antwort in einer Zeile ist nicht zulässig.

Antworten des Servers werden immer mit einem Timestamp eingeleitet. Einzige Ausnahme ist der Welcome Der Timestamp wird, sofern nicht anders definiert, aus der aktuellen Systemzeit wie folgt gebildet: Angabe der Sekunden seit dem 1.1.1970 00:00:00 (POSIX-Sekunden") einem Punkt und den Millisekunden der laufenden Sekunde als Festkommazahl mit 3 Nachkommastellen.

In diesem Text wird bei allen Angaben der Zeitstempel implizit immer vorangestellt, aus Gründen der Übersichtlichkeit jedoch nicht angegeben, sofern es nicht zur Verdeutlichung eines Sachverhaltes notwendig ist.

Nachdem ein Server die Antwort gesendet hat, bearbeitet er das nächste Kommando. Kommandos werden so in einer wechselseitigen Folge von Befehlen und deren Reaktion durch eine streng wechselseitige Kommunikation zwischen Client und Server abgewickelt. Der Server MUSS die Kommandos in der zeitlichen Reihenfolge des Eintreffens verarbeiten.

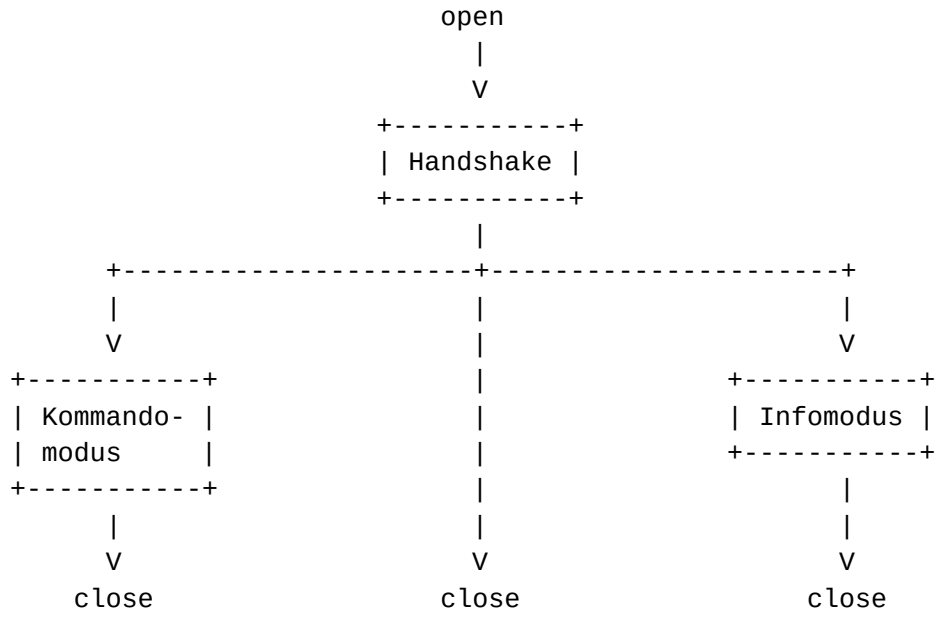
Ist das Ergebnis eines Kommandos sowohl aufgrund einer Kommunikation mit angeschlossenen Modellbahnelementen, als auch anhand sonstiger Daten ermittelbar, so SOLL die Möglichkeit der direkten Anlagenabfrage benutzt werden. Stehen für eine Anfrage keine ausreichend abgesicherten Ergebnisse bereit, so ist die Antwort `416 ERROR no data`.

3 Serverzustand

Ein SRCP Server befindet sich gegenüber dem Client immer in einem von 3 möglichen Zuständen: Handshake, Kommandomodus oder Infomodus.

Mit der Verbindungsaufnahme wird der Handshakemodus aktiviert. In diesem werden vom Client die weiteren Betriebsparameter festgelegt. Aus dem Handshakemodus wechselt die Verbindung auf Anforderung des Clients entweder in

den Kommandomodus oder den Infomodus. Aus allen drei Modi kann die Verbindung jederzeit geschlossen werden, ein expliziter Endebefehl ist nicht spezifiziert.



4 Kommandos

Im weiteren werden die SRCP Kommandos und ihre jeweiligen Antworten detailiert beschrieben. Sie müssen alle von einem SRCP Server implementiert werden.

Kommandos werden ausschließlich während des Handshakes und im Kommandomodus abgewickelt. Im Infomodus sind sie nicht zulässig und MÜSSEN ignoriert werden.

Kommandos MÜSSEN in der Reihenfolge ihres Eintreffens vom Server abgearbeitet werden. Insbesondere betrifft dies die Reihenfolge bei der Übermittlung an Anlagenbestandteile.

Unbekannte oder nicht erkannte Kommandos MÜSSEN mit der Meldung `410 ERROR unknown command` quittiert werden.

4.1 Kommandos im Handshake

Ein Client stellt eine TCP/IP Verbindung zum Server her. Der Server sendet einen einzeiligen Welcomestring. Daraufhin werden zwischen Client und Server die Kommunikationsparameter und der gewünschte Betriebsmodus ausgehandelt. Durch den abschließenden Befehl `GO`, der vom Client an den Server gesendet wird und von diesem bestätigt wird, startet der Betriebsmodus.

Der Server MUSS intern eine Unterscheidung der verschiedenen Clientsessions vornehmen. Eine einmal vergebenes Kennzeichen DARF während der gesamten Laufzeit des Servers NICHT ein zweites Mal verwendet werden.

4.1.1 Welcome

Nachdem eine Verbindung zwischen Client und Server hergestellt wurde, sendet der Server eine Textzeile an den Client: den Welcome. Dieser besteht aus durch das Zeichen ";"(Semikolon, #59) zu trennenden Informationen über den Server. Jede dieser Informationen ist als SSchlüssel "WertPaar anzusehen. Die Reihenfolge der Schlüssel ist nicht festgelegt. Ebenso ist es im allgemeinen zulässig, dass Schlüssel mit unterschiedlichen Werten wiederholt angegeben

werden. Pro Schlüssel ist genau ein Wert zulässig. Ein Wert besteht aus einem oder mehr durch Whitespace getrennten Worten. Das Zeilenende gilt als Ende des letzten Wertes, ein Semikolon DARF fehlen.

Der Welcome DARF bei nicht mehr ausreichenden Ressourcen als Fehlermeldung gestaltet werden. In diesem Fall MUSS der Server nach dem Senden des Welcomes die Verbindung zum Client schließen und DARF KEINE Kommandos des Clients entgegennehmen und ausführen: `500 ERROR out of ressources`

Folgende Schlüssel MÜSSEN im normalen Welcome angegeben werden:

- SRCP <Version>: Vom Server unterstützte SRCP Version. Sie MUSS exakt einer der freigegebenen Versionsnummern entsprechen. Diese Angabe MUSS genau einmal angegeben werden. Eine mehrfache Angabe ist nicht zulässig.

Folgende Schlüssel KÖNNEN angebeben werden, um definierte Informationen anzuzeigen. Einem Server steht es frei, weitere Schlüssel zu verwenden. Schlüssel, deren Bezeichner mit der Zeichenfolge SRCP beginnt, DÜRFEN NICHT verwendet werden, wenn sie nicht im SRCP definiert sind.

- SRCPOTHER <Version>: Vom Server zusätzlich unterstützt SRCP Version. Sie MUSS exakt mit einer freigegebenen Versionsnummer entsprechen und von der Angabe im Schlüssel `SRCP` verschieden sein.

4.1.2 Handshake

Nachdem die Verbindung hergestellt und der Welcome gesendet wurde, wartet der Server auf ein Kommando des Clients. Dieses führt der Server aus und generiert eine einzeilige Antwort, die an den Client gesendet wird. Im Anschluß daran wartet der Server auf das nächste Kommando. Auf diese Weise werden wechselseitig Informationen und Kommandos ausgetauscht.

Folgende Kommandos sind definiert, werden sie nicht ausgeführt, gelten die angegebenen Standardeinstellungen. Andere als die Aufgeführten sind in der Handshakephase verboten.

SET PROTOCOL SRCP <VERSION> Das gewünschte SRCP Protokoll, das benutzt werden soll, wird vereinbart. Fehlt dieses Kommando wird die SRCP Version voreingestellt, die im Welcome angegeben wurde. Der Server sendet entweder ein `201 OK PROTOCOL SRCP` oder eine Fehlermeldung `400 ERROR unsupported protocol` als Antwort.

SET CONNECTIONMODE SRCP <MODE> Die Art der Verbindung wird vereinbart. Zulässig sind INFO für den unidirektionalen Informationsbetrieb und COMMAND für den bidirektionalen Kommandobetrieb. Der Server sendet entweder ein `202 OK CONNECTIONMODE` oder eine Fehlermeldung `401 ERROR unsupported connection mode` an den Client. Fehlt dieses Kommando wird als connection mode der COMMAND Modus gesetzt.

GO Durch diesen Befehl wird die Handshakephase beendet und der jeweilige Betriebsmodus aktiviert. Der Server sendet als Quittung `200 OK GO <ID>` an den Client. Wenn der Befehl nicht ausgeführt werden kann, so sendet der Server die Fehlermeldung `402 ERROR unsufficient data` und verbleibt im Handshakemodus. Dies ist für zukünftige Erweiterungen und das Einbetten von SRCP in andere Protokolle vorgesehen. Das Feld <ID> kennzeichnet die vom Server vergebene numerische Session-ID. Diese ist für den Server eindeutig und wird niemals, solange der Server läuft, zweimal vergeben. Sie DARF NICHT nicht identisch mit Null sein.

Nach Abschluß des Handshakes können die dort vereinbarten Eigenschaften nicht mehr verändert werden und gelten für die gesamte weitere Verbindung.

Folgende Fehlermeldungen sind im Handshake zulässig:

`400 ERROR unsupported protocol` Protokoll wird nicht unterstützt.

`401 ERROR unsupported connection mode` Verbindungsmodus wird nicht unterstützt.

`402 ERROR unsufficient data` Ungenügende Angaben.

`410 ERROR unknown command` Unbekanntes Kommando.

`500 ERROR out of ressources` Keine Ressourcen mehr frei.

4.2 Kommandomodus

Im Kommandomodus wartet der Server auf Kommandos des Clients und führt diese aus. Im Ergebnis der Ausführung generiert der Server immer eine Antwort, die er auf dem gleichen TCP Session an den Client zurücksendet. Anschließend wird das nächste Kommando ausgeführt. Eine überlappende oder sonstig von der Reihenfolge des Eintreffens abweichende Bearbeitung ist nicht zulässig.

Die Antworten des Servers an den Client im Kommandomodus beginnen immer mit einem Zeitstempel, gefolgt von einem numerischen Antwortcode und weiteren Angaben. Der numerische Antwortcode ist in Gruppen gegliedert

1xx INFO: Informationen, Ergebnisse

2xx OK: Befehl wurde angenommen und zur Ausführung gebracht. Achtung: Dies ist keine Bestätigung, das der Befehl auch tatsächlich ausgeführt wurde.

4xx/5xx ERROR: Eine Fehlerbedingung ist aufgetreten, der Befehl wird ignoriert und nicht ausgeführt.

6xx ERROR: Ein serverspezifischer Fehlerzustand ist aufgetreten. Der betreffende Befehl wird nicht ausgeführt. Details sind der Serverdokumentation zu entnehmen.

Die Bedeutung des Zeitstempels ergibt sich der Art der Antwort: Für Quittungen (OK, ERROR) ist er der Zeitpunkt, an dem das das Kommando abgearbeitet und die Quittung (OK oder ERROR) generiert wurde. Bei einem INFO kennzeichnet der Zeitstempel denjenigen Zeitpunkt, an dem das gemeldete Ereignis dem SRCP Server erstmalig bekannt wurde, oder, sofern der Server dies ermitteln konnte, zu dem das Ereignis eingetreten ist. Wenn es sich nur um eine teilweise Aktualisierung von Parametern einer Parameterliste handelt, so ist dies der Zeitpunkt der letzten Änderung.

4.2.1 Kommandos

Gültige Kommandos MÜSSEN immer ausgeführt werden. Eine Überprüfung des Kommandos mit dem aktuellen Wissen des Servers mit dem Ziel, Kommandos zu unterdrücken, die z.B. keine Veränderung bewirken könnten, ist nicht zulässig (Optimierungsverbot). Werden von verschiedenen Sessions Kommandos für ein und dasselbe Gerät gesendet, so sind diese in der Reihenfolge des Eintreffens auszuführen.

Folgende Kommandos sind definiert:

GET Abfrage von Werten

SET Setzen von Werten

CHECK Prüft einen Befehl

WAIT Wartet auf Werte

INIT Initialisierung von Elementen

TERM Beendet von mit INIT initialisierte Elemente

RESET Reinitialisiert ein Element

VERIFY Überprüft die Einstellungen eines Elements

Das Kommando INIT dient generell der Initialisierung und Konfiguration des angegebenen Gerätes bzw. der Gerätegruppe. Nach einem INIT befindet sich das betreffende Element im Grundzustand. Im Infomodus wird für das betreffende Element die entsprechende `101 INFO`-Meldung gesendet, sofern definiert.

Der INIT informiert den Server über die Eigenschaften des betreffenden Gerätes, die daraufhin modifiziert werden KÖNNEN. Der Befehl verändert weder die Buszugehörigkeit noch die Adresse.

Nach einem INIT ist für alle aktiven Sessions jeglicher Schreibzugriff mittels SET, RESET oder INIT auf das betreffende Gerät unzulässig, bis ein Lesezugriff durch die Befehle GET, WAIT oder VERIFY (je nach Gerätegruppe) oder ein TERM ausgeführt wurde. Einzige Ausnahme hiervon ist die Session, die den INIT veranlaßt hat.

Durch das Kommando TERM wird die Initialisierung wieder aufgehoben und das betreffende Element in den Grundzustand versetzt. Anschließend kann keine Auskunft über das Element gegeben werden. Vor einer erneuten Benutzung muß das betreffende Element neu initialisiert werden. Im Infomodus wird für das betreffende Element die entsprechende `102 INFO`-Meldung gesendet.

Durch den TERM eines Gerätes wird auch ein eventueller LOCK ohne weitere Befehle beendet (impliziter TERM LOCK).

Der Befehl SET verändert bestimmte Parameter des betreffenden Elements. Durch den SET Befehl KANN das Element implizit initialisiert werden. Die Antwort auf einen SET ist im Erfolgsfall ein `200 OK`. Der Befehl CHECK entspricht in allen Aspekten dem Befehl SET, nur dass die tatsächliche Ausführung des Befehls unterdrückt wird. Zweck dieses CHECK-Befehls ist die Unterstützung des Debuggings von Clients.

Der Befehl GET ermittelt die aktuellen Parameter. Dies umfaßt auch programmierte Parameter, die nur über ein INIT eingestellt werden können. Ergebnis ist die `100 INFO` oder eine geeignete Fehlermeldung. Die Informationen stellen "bestes Wissen"dar. Wenn ein Server die Angaben durch Abfragen der Anlagenkomponenten ermitteln kann, so SOLL er diese abfragen. Wenn die aktuelle Situation oder das zugrundeliegende System es nicht zuläßt, so KANN er auf eigene Informationen, die z.B. aus dem letzten SET resultieren, zurückgreifen. In diesem Fall ist der numerische Code nicht 100 sondern 110, also `110 INFO...`. In keinem Fall sind die Informationen als garantiert identisch mit dem tatsächlichem Anlagenzustand anzusehen.

Der Befehl WAIT wartet eine begrenzte Zeitspanne auf das Eintreffen eines bestimmten Musters im Zustand eines Gerätes. Dieses Warten blockiert die Clientsession. Wird diese Zeitspanne überschritten, wird eine Fehlermeldung `417 ERROR timeout` generiert. Trifft der erwartete Zustand in der angegebenen Zeit ein, wird ein `200 OK` generiert.

Der Befehl RESET dient dem erneuten Initialisieren eines Elements. Dieser Befehl SOLL eine verkürzte Abfolge von TERM/INIT sein. Das Ergebnis ist entweder ein `200 OK` im Erfolgsfall oder eine Fehlermeldung. Das betreffende Element MUSS sich anschließend im Grundzustand befinden.

Der Befehl VERIFY dient der Überprüfung, ob ein Elementzustand mit einem vorgegeben Muster übereinstimmt. Das Ergebnis ist entweder ein `200 OK` wenn dies zutrifft oder eine Fehlermeldung. Dies ist insbesondere bei programmierbaren Geräten anzuwenden, wenn die betreffenden Parameter nicht durch ein SET beeinflußbar sind oder nicht via GET direkt ermittelbar sind.

4.2.2 Kommandoparameterliste und Adressierung

Parameter der Kommandos sind als Liste anzugeben. Einzelne Parameter werden durch Whitespace voneinander getrennt. Die Position des Parameters innerhalb der Liste kennzeichnet seine Bedeutung (positionsbezogene Parametrierung). Ein einzelner Parameter DARF KEINEN Whitespace enthalten. Für einzelne Befehle können mehrere unterschiedliche Parameterlisten definiert sein, die sich anhand der Anzahl der Parameter unterscheiden und eine unterschiedliche Bedeutung haben.

Die Adressierung der Elemente einer Anlage basiert auf den folgenden Elementen: dem Bus, der Gerätegruppe und, sofern mehrere Geräte möglich sind, der Geräteadresse.

Ein Bus ist grundsätzlich die Abstraktion eines konkreten Hardwarekommunikationsstrangs. Ein Bus ist z.B. durch eine Central Unit repräsentiert. Ein Bus SOLL auch genutzt werden, um parallele Adreßbereiche auf einem realen System abzubilden.

Ein Bus wird durch die Konfiguration eines Servers festgelegt und initialisiert. Im laufenden Betrieb ist eine Veränderung an den Bussen nur über den Reset eines Servers zulässig.

Ein Bus wird durch eine laufende Nummer bezeichnet, die bei Null (0) beginnt und lückenlos aufsteigend gezählt wird. Bus 0 ist für den Server selbst reserviert.

Die Busnummer ist bei allen Kommandos als erster Parameter unmittelbar nach dem Kommandowort anzugeben und wird bei allen Antworten und Informationen verwendet.

Gerätegruppen sind Gegenstand des nachfolgenden Abschnitts

Zielpunkt der Adressierung sind die Geräte: De- bzw. Encoder (im weiteren wird nur von Decodern gesprochen, Encoder gelten sinngemäß). Jeder Decoder ist Element mindestens eines Busses und einer Gerätegruppe, die gleichzeitige Zugehörigkeit eines Decoders zu mehreren Bussen u/o Gerätegruppen ist zulässig.

Wechselt die Buszugehörigheit (z.B. bei Loks), so werden die Daten nicht automatisch übernommen. Es ist Aufgabe der Clients, diesen Transfer zu erkennen und darauf zu reagieren.

Wenn auf einem Bus Geräte mit einer identischen Hardwareadresse jedoch unterschiedlichem Ansteuerungsprotokoll koexistieren, so MUSS der SRCP Server sicherstellen, das das betreffende Gerät nachvollziehbar agiert.

Bei der Angabe von Parametern sind Wildcards im allgemeinen nicht zulässig. Bei einzelnen Gerätegruppen KÖNNEN sie entsprechend der folgende Bedeutung unterstützt werden:

- * (Asterisk, #42): Alle dem Server bekannten Werte. Sind dem SRCP Server keine bekannt, wird eine Fehlermeldung `416 ERROR no data` erzeugt.
- = (Gleichheitszeichen, #61): Dieser Wert wird nicht verändert. Kennt der SRCP Server den aktuellen Wert nicht, wird eine Fehlermeldung `416 ERROR no data` erzeugt.

Für einen SRCP Server gelten alle diejenigen Geräte als bekannt, die bereits angesprochen wurden und nicht mittels TERM gelöscht wurden. Zusätzlich KÖNNEN Elemente von der Anlage selbst gemeldet werden, sofern diese dies ermöglicht. Hierbei KANN auch ein Gerät im Grundzustand als bekannt gelten. Bei einigen Hardwaresystemen werden einige Aktivitäten durch bestimmte Eigenschaften der übermittelten Daten abgebildet. Ein SRCP Server KANN diese selbst einsetzen, um das definierte Wildcardverhalten umzusetzen. Bei Systemen, die dies nicht direkt unterstützen, MUSS der Server diese Funktion emulieren. Ein SRCP Server DARF eventuell mögliche direkte Parametrierungen dieser Art NICHT von einem Client annehmen. Dies betrifft insbesondere speziell reservierte Adressen (z.B. für Broadcasts).

4.2.3 Fehlermeldungen

Fehlermeldungen werden vom Server als Antwort auf Kommandos des Clients generiert, wenn das Kommando nicht wie verlangt ausgeführt werden kann.

Fehlermeldungen beginnen wie alle Antworten immer mit dem Timestamp, gefolgt von dem dem Fehlercode und der Angabe `ERROR`. Daran MUSS sich eine der in der nachfolgenden Liste angegebenen Meldungen anschließen:

`410 ERROR unknown command` Unbekanntes Kommando.

`411 ERROR unknown value` Unbekannter Wert.

`412 ERROR wrong value` Ein Parameter ist außerhalb des zulässigen Bereichs.

`413 ERROR temporarily prohibited` Befehl ist derzeit verboten.

`414 ERROR device locked` Gerät ist gesperrt.

`415 ERROR forbidden` Befehl ist grundsätzlich verboten.

`416 ERROR no data` Keine Informationen verfügbar.

`417 ERROR timeout` Ein Zeitlimit ist überschritten.

`418 ERROR list to long` Die Parameterliste ist zu lang.

`419 ERROR list to short` Die Parameterliste ist zu kurz.

`420 ERROR unsupported device protocol` Dieses Gerät unterstützt das angegebene Protokoll nicht.

`421 ERROR unsupported device` Dieses Gerät wird auf dem angegeben Bus nicht unterstützt.

`422 ERROR unsupported device group` Die Gerätegruppe ist auf dem angegebenen Bus nicht verfügbar.

`423 ERROR unsupported operation` Die angeforderte Aktion/Befehl wird von diesem Gerät nicht unterstützt.

`424 ERROR device reinitialized` Das betreffende Gerät wurde neu initialisiert. Ein GET Befehl muß zunächst ausgeführt werden.

`499 ERROR unspecified error` Ein Fehler ist aufgetreten, kann aber nicht näher angegeben werden. Diese Fehlermeldung SOLL vermieden werden.

Wenn es weitere Informationen zu dem betreffenden Fehler gibt, so können diese im Anschluß an die obige Meldung angehängt werden. Ein Client KANN diese Angaben ignorieren.

4.2.4 Gerätegruppen

Gerätegruppen sind Zusammenstellungen von gleichartigen Geräten. Folgende Gerätegruppen sind definiert:

GL Generic Loco

GA Generic Accessoire

FB Feed Back

SM Service Mode

TIME Zeitnormal

POWER Energieversorgung

SERVER SRCP Server

SESSION SRCP Clientsession

LOCK Geräte, die andere Geräte sperren

DESCRIPTION Geräte, die andere Geräte und Gerätegruppen beschreiben

Einige Gerätegruppen entsprechen Anlagenelementen. Diese werden im Fall von mehreren unterschiedlichen Betriebsmodi per default auf den kleinsten gemeinsamen Nenner initialisiert. Eine Nutzung der weitergehenden Features erfordert eine Initialisierung, bei der hardwarenahme Informationen benötigt werden.

Jeder SRCP Server MUSS die Gerätegruppen SERVER, SESSION und DESCRIPTION im Bus 0 unterstützen. Für jeden weiteren Bus MÜSSEN die Gerätegruppen POWER und DESCRIPTION verfügbar sein. Alle anderen Gerätegruppen sind optional. Werden sie unterstützt, MUSS dies in der DESCRIPTION des Busses angegeben werden. Eine Unterstützung der DESCRIPTION für Gerätegruppen ist im Unterschied zu den DESCRIPTION der Busse optional und MUSS aber für jede Gerätegruppe des betreffenden Busses ggf. vorhanden sein.

Die Gerätegruppe GA ist für Decoder vorgesehen, die unter einer Adresse einen oder mehr Ports mit jeweils zwei oder mehr möglichen Zuständen bereitstellen. Dies sind üblicherweise ortsfeste Anlagenelemente wie Weichen und Entkuppler. Die Gerätegruppe GL kennzeichnet Lokdecoder: Dekoder mit mind. einem Ausgang für die Fahrstufen, einem für die Fahrtrichtung und ggf. Funktionsausgängen. Feedback Geräte FB sind Encoder, die das Auftreten eines Ereignisses auf der Anlage signalisieren. Sie haben exakt einen Eingang, der vermittels einer Adresse identifiziert wird und der über mindestens zwei unterscheidbare Zustände verfügt. Sind einzelne Encoder zu einer Gerätegruppe zusammengefaßt, so hat dies keine Auswirkungen auf die Identifikation.

Der Service Mode SM betrifft Decoder, die im Programmiermodus betrieben werden. Hierfür KANN es erforderlich sein, das der Decoder auf einen anderen Bus versetzt werden muß (Programmiergleis) oder ein Bus für andere Befehle gesperrt wird. In diesem Bus KÖNNEN normale Steuerbefehle dann mit der Fehlermeldung `temporarily prohibited` oder `forbidden` abgewiesen werden. Eine technische Realisierung KANN einen direkten Durchgriff auf das benutzte Digitalsystem umfassen.

Das Zeitnormal TIME ist nur in der Busnummer 0 zulässig. Es dient der Bereitstellung einer einheitlichen Modellzeit. Diese wird ausgehend von einem Startzeitpunkt schneller oder langsamer als die Echtzeit aber genau wie diese monoton mit Modellsekundengenauigkeit gezählt. Das Zeitnormal ist optional. Es ist nur ein Gerät in dieser Gerätegruppe zulässig, eine Adressierung entfällt.

Die LOCK Geräte sind Geräte, die eine Sperre über ein anderes Gerät bereitstellen. Sie sind optional. Wenn vorhanden, so MUSS der Server vor Aktionen auf den lockbaren Geräten den LOCK konsultieren. Sie DÜRFEN auf einzelnen Bussen existieren, und auf anderen Bussen nicht. Es ist jedoch anzustreben, das alle Busse eines Servers mit Lockgeräten versehen sind. Durch einen LOCK sind die folgenden Befehle auf dem Gerät exklusiv für die den LOCK haltende Session reserviert: SET, TERM, INIT, RESET.

Die DESCRIPTION Geräte können andere Geräte mit ihren Eigenschaften beschreiben. Es wird je nach Art der Parameterliste zwischen der DESCRIPTION von Bussen und der DESCRIPTION von Gerätegruppen unterschieden.

POWER kennzeichnet den Zustand der Energieversorung des betreffenden Busses. POWER im Bus 0 ist nicht zulässig. Das Aktivieren der Energieversorgung in einem Bus kann implizit auch andere Busse aktivieren, dies MUSS im INFO Modus berücksichtigt werden!

SERVER kennzeichnet zusammen mit dem Bus 0 Aktivitäten, die den Server betreffen (insb. TERM und RESET). Eine Verwendung mit anderen Bussen ist nicht zulässig. Der SRCP Server ist einziges Element dieser Gerätegruppe, eine weitere Adressierung entfällt.

SESSION kennzeichnet zusammen mit dem Bus 0 die Menge der aktiven Clientverbindungen. Eine Verwendung anderer Busse als 0 ist nicht zulässig.

Folgende Übersicht soll die zulässige Verteilung der Gerätegruppen und Befehle auf die Busse darstellen. Nicht jedes Gerät ist auf jedem Bus zulässig:

	SET	GET	WAIT	INIT	TERM	RESET	VERIFY
	CHECK						
GL	1..	1..	--	1..	1..	1..	--
SM	1..	1..	--	1..	1..	1..	1..
GA	1..	1..	--	1..	1..	1..	--
FB	--	1..	1..	1..	1..	1..	--
TIME	0	0	0	0	0	0	--
POWER	1..	1..	--	1..	1..	--	--
SERVER	--	0	--	--	0	0	--
SESSION	--	0	--	--	0	--	--
LOCK	0..	0..	--	0..	0..	--	--
DESCRIPTION	--	0..	--	--	--	--	--

Die mit - gekennzeichneten Einträge markieren Kombinationen, die NICHT verwendet werden DÜRFEN.

Gerätegruppe FB: Feedback Rückmelder haben eine Adresse, unter der genau ein aktueller Wert erfaßt ist. Im einfachsten Fall ist ein Rückmelder ein binäres Signal. Ein vom Server beeinflußbarer Grundzustand ist nicht vorhanden.

INIT

```
INIT <bus> FB <optionale Parameter zur Initalisierung>
```

Ein serverseitig vorkonfiguriertes Rückmeldesystem wird initialisiert. Die exakten Parameter sind nur dem Server bekannt. Eine Initialisierung einzelner FB Geräte ist nicht vorgesehen. Die optionalen Parameter KÖNNEN serverseitige Vorgaben verändern.

GET

```
GET <bus> FB <addr>
```

Liefert den aktuellen Wert des Rückmelders als `100 INFO`.

TERM

```
TERM <bus> FB
```

Mit dem TERM werden auf dem betreffendem Bus alle FB aus der laufenden Kommunikation herausgenommen, Die angeschlossene Hardware SOLL hiermit abgeschaltet werden, wenn alle Busse, die darauf zugreifen, terminiert wurden. Laufende WAIT MÜSSEN mit timeout beendet werden, sofern die WAIT Bedingung nicht erfüllt ist.

WAIT

```
WAIT <bus> FB <addr> <value> <timeout>
```

Wartet bis zu <timeout> Sekunden (Echtzeit) auf das Erreichen des spezifizierten Wertes. Wenn das Wert bereits vorliegt, so wird nicht gewartet, sondern dessen Eintreten sofort gemeldet.

INFO

```
100 INFO <bus> FB <addr> <value>
101 INFO <bus> FB
102 INFO <bus> FB
```

Der aktuelle Wert des Rückmelders bzw. die Initialisierung resp Terminierung des FB-Systems auf diesem Bus.

Gerätegruppe GA: Generic Accessoire Ein Generic Accessoire kennzeichnet allgemein einen Decoder, der unter einer Adresse einen oder mehrere Ausgänge (Ports) bedienen kann. Dies sind häufig Weichendecoder oder Signaldecoder, die als Impulsdecoder arbeiten. Hier ist zu beachten, dass es Einschränkungen bei der gleichzeitigen Aktivierung und/oder Deaktivierung von Ausgängen geben kann. Diese sind ggf. der Beschreibung des Decoders zu entnehmen. Ein SRCP Server kann diese Eigenschaften nicht immer selbst erkennen und melden!

Der Grundzustand eines Gerätes ist durch Null auf allen Ports gekennzeichnet.

INIT

```
INIT <bus> GA <addr> <device protocol> <optional weitere Parameter>
```

Mit diesem Befehl wird ein GA Gerät im Server initialisiert. Folgende Parameter sind zulässig:

addr Adresse aus dem Protokollbereich.

device protocol

- M Märklin/Motorola-Format: addr 1..256, port: 0,1, value: 0,1
- N NMRA-DCC-Format: addr 1..511, port: 0,1, value: 0,1
- P Protocol by Server: Der Server bestimmt den Protokolltyp anhand seiner Konfiguration. Addr, port und value nicht beschränkt.

SET

```
SET <bus> GA <addr> <port> <value> <delay>
```

Der Port <port> des Decoders mit der Adresse <addr> wird für <delay> Millisekunden auf den Wert <value> gesetzt. Nach Ablauf der Zeit wird vom Server automatisch der Wert 0 an den Decoder gesendet. Ist die Verzögerung -1, unterbleibt die automatische Deaktivierung und der Ausgang bleibt solange aktiv, bis er von einem weiteren SET Befehl deaktiviert wird. Ein SRCP Server KANN während dieser Zeit kurzzeitige De- und Reaktivierung des Ports ausführen. Dies ist ggf. ein Erfordernis der Hardware, die ein Überschreiten bestimmter maximaler Einschaltdauern nicht zuläßt. Ein Delay von 0 ist nicht zulässig.

Bedeutung der Argumente

addr Zahl > 0 (Nummer der Weiche/des Signals)

port 0, 1, 2,... (Ausgang des Decoders), wird ein nicht unterstützter Port angegeben, wird eine Fehlermeldung `412 ERROR wrong value` erzeugt.

value 0, 1, 2, ... (0 => deaktiviert, >0 => aktiviert), wird ein ungültiger Wert angegeben, wird eine Fehlermeldung `412 ERROR wrong value` erzeugt.

delay Wert in Millisekunden. Gibt an, nach welcher Zeit der Server einen aktivierten Port automatisch deaktivieren (d.h. auf 0 setzen) soll. Wird 1äls delay übergeben, dann wird der Port nicht automatisch deaktiviert. Ist action=0 (Deaktivierung) wird delay ignoriert, muß aber angegeben werden. Ein Port gilt als aktiv, wenn sein Status ungleich Null ist. Ein Delay von 0 ist für Values <0 nicht zulässig, es wird eine Fehlermeldung `412 ERROR wrong value` erzeugt.

GET

```
GET <bus> GA <addr> <port>
```

Der aktuelle Zustand des Ports wird an den Client übermittelt.

INFO

```
100 INFO <bus> GA <addr> <port> <value>
101 INFO <bus> GA <addr> <device protocol> <optional weitere Parameter>
102 INFO <bus> GA <addr>
```

Das Gerät an der angegebenen Adresse hat am angegebenen Port den angegebenen Wert bzw die Initialisierungs und Terminierungsangaben.

Gerätegruppe GL: Generic Loco Ein Generic Loco kennzeichnet allgemein alle Lokdecoder.

Der Grundzustand ist Null bei allen Parametern.

INIT

```
INIT <bus> GL <addr> <device protocol> <optional weitere Parameter>
```

Mit dem INIT wird die Art und grundlegende Eigenschaften des Decoders dem Server bekannt gemacht. Eine Veränderung der Decodereigenschaften bei programmierbaren Decodern ist nicht zulässig (hierfür ist die Gerätegruppe SM zu verwenden), Ausnahme: Wenn sich Decoder durch eine Sequenz normalerSSteuerbefehle programmieren lassen, so DARF NICHT die SM Gerätegruppe benutzt werden.

Zulässige Werte der Parameter

device protocol

nachfolgend sind die derzeit gültigen Werte definiert.

- `A` Analogbetrieb addr: 0, keine f (Zusatzinfo: keine Fahrstufen)
- `F` Reserviert für Fleischmann
- `L` Reserviert für Loconet
- `M` Reserviert für Märklin/Motorola

  `<Protokollversion> <Dekoderfahrstufen> <Anzahl_Dekoderfunktionen>`

  - **Protokollversion 1** ist für das älteMM Protokoll (z.B. Delta Dekoder)
  - **Protokollversion 2** ist für das neueProtokoll: zusätzlich 4 Funktionen, 14, 27 oder 28 Fahrstufen, maximal 256 Adressen)

  Die Dekoderfahrstufen entsprechen denen, die der Dekoder tatsächlich unterstützt (ggf. unter Beachtung der Anzahl der Dekoderfunktionen). Die Anzahl der Dekoderfunktionen ergibt sich aus der Anzahl der ansprechbaren Funktionen, ohne Beachtung einer evt. nicht genutzten Funktion. Sie sind immer fortlaufend ohne Unterbrechung anzusehen.
- `N` Reserviert für NMRA/DCC:

  `<Protokollversion> <Dekoderfahrstufen> <Anzahl_Dekoderfunktionen>`

  - **Protokollversion 1** ist für das "kurzeÄdreßformat
  - **Protokollversion 2** ist für das langeÄdreßformat

  Die Ausführungen zu den Dekoderfahrstufen und Anzahl Funktionen gelten wie bei `M`.
- `P` protocol by server: beliebige Adresse, beliebige Anzahl Funktionen (Zusatzinfo: beliebige Anzahl Fahrstufen). Der Server kennt den Typ des Decoders.
- `S` Reserviert für native Selectrix
- `Z` Reserviert für native Zimo

Ein Server KANN ein oder mehr dieser Angaben nicht unterstützen. Nicht angeführte Angaben DÜRFEN NICHT verwendet werden. Diese Angaben werden in zukünftigen Revisionen des SRCP konkretisiert.

Hinweis: Bei einigen Protokollen existieren mehrere Bereiche für Funktionen, üblich sind eine fahrtrichtungsabhängige und mehrere von der Fahrtrichtung unabhängige. Ein Server MUSS die fahrtrichtungsabhängigen vor den fahrtrichtungsunabhängigen anordnen.

GET

```
GET <bus> GL <addr>
```

Die Antwort ist eine `100 INFO`-Meldung.

SET

```
SET <bus> GL <addr> <drivemode> <V> <V_max> <f1> .. <fn>
```

Der Decoder wird mit den angegeben Werten belegt. Der Grundzustand ist durch den Wert 0 für alle Parameter (außer der Adresse) gekennzeichnet. Die Bedeutung der Parameter ist allg. folgende:

addr Zahl >= 0 oder * (Asterisk, #42), die am Decoder eingestellte Adresse, entsprechend dem Protocol können Einschränkungen gelten. Der * kennzeichnet die Broadcastadresse, bei der alle dem SRCP Server bekannten Loks auf die angegebenen Werte gesetzt werden. Wenn eine Adresse eine Sonderfunktion im zugrundeliegenden System hat (Broadcast), so DARF sie NICHT akzeptiert werden und MUSS mit einer Fehlermeldung abgewiesen werden.

drivemode

- `=` unverändert
- `0` rückwärts
- `1` vorwärts
- `2` Nothalt

Wird ein nicht unterstützter Fahrmodus angegeben, wird eine Fehlermeldung `412 ERROR wrong value` erzeugt.

V, V_max Fahrstufe und maximale Fahrstufe, 0 ist Stillstand, ein Wert >0 bedeutet Bewegung der Lok (d.h. eine reale Fahrstufe >0), V<0 ist nicht zulässig ebenso V>V_max. Die Geschwindigkeit steigt von 0 bis V_max, solcherart, daß eine Geschwindigkeit V2>V1 bewirkt, das die Lok bei V2 nicht langsamer als bei V1 fährt. In allen Fehlersituationen wird eine Fehlermeldung `412 ERROR wrong value` erzeugt. Der Einsatz des Wildcard = ist nur bei beiden Parametern gleichzeitig zulässig.

f1 .. fn = (= unverändert), 0 (= aus), 1 (= an) nicht zulässige Werte bewirken eine Fehlermeldung `412 ERROR wrong value`.

Ein Beispiel:

```
INIT 1 GL 1 N 1 128 5
SET 1 GL 1 1 4 100 1 0 1 0 =
```

Die über den Bus 1 adressierte Lok 1 (NMRA Decoder, 128 Dekoderfahrstufen, 1+4 Funktionen) fährt mit Speedstep 4 (von 100 möglichen) vorwärts. Von den 5 Funktionen sind FUNKTION und F2 aktiviert. Der Wert von F4 wird anhand des Wissens des Servers über den aktuellen Zustand gesetzt.

INFO Der INFO MUSS alle aktuellen Werte umfassen, unabhängig ob sie aktuell geändert wurden oder nicht. Das Format ist wie folgt:

```
100 INFO <bus> GL <addr> <drivemode> <V> <V_max> <f1> .. <fn>
101 INFO <bus> GL <addr> <device protocol> <optional weitere Parameter>
102 INFO <bus> GL <addr>
```

Hierbei ist der Parameter V nicht der vom Client übermittelte Speedstep, sondern der vom Server an die Lok gesendete, echte Fahrstufe unter Auslassung eventueller Sonderbedeutungen: Die Fahrstufe 0 kennzeichnet Stillstand, die Fahrstufen von 1 bis einschließlich V_max kennzeichnen die Bewegung der Lok im angegebenen <drivemode>.

TERM

```
TERM <bus> GL <addr>
```

Setzt die Lok in den Grundzustand und entfernt die Lok aus dem Wissenbestand des Servers.

Gerätegruppe SM: Servicemode Die Geräte des Servicemode werden für die permanente Veränderung von Geräten bereitgestellt. Sie sind optionaler Bestandteil. Werden sie nicht unterstützt, so MUSS eine Fehlermeldung `422 ERROR unsupported device group` generiert werden. Sie sind für den Einsatz zur Decoderprogrammierung vorgesehen

Ein Grundzustand ist nicht definiert.

INIT

```
INIT <bus> SM <protocol>
```

Initialisiert den betreffenden Bus für die Programmierung von Decodern im angegebenen Protokoll. Als Folge hiervon KANN der Bus für andere Aufgaben gesperrt sein.

Zulässige Protokolle sind

NMRA: Decoder nach NMRA Standard.

SET

```
SET <bus> SM <decoderaddress> <type> <1 or more values>
```

Folgende <more values> sind definiert.

type

- **CV**: Configuration Variable: der erste Parameter der Restliste kennzeichnet die Adresse einer NMRA Decodervariablen (<typeaddress>). Diese erhält den mit dem zweiten Parameter <value> beschriebenen Wert. Einschränkungen im Wertebereich der Parameter ergeben sich aus der Spezifikation des Decoders. Ein Server kann Decoderspezifische Wertebereiche nicht überprüfen.
- **CVBIT**: Configuration Variable Bit: 3 Parameter: <typeadress> <bit> <value> Das Bit <bit> der Adresse <typeadress> wird auf den Wert <value> gesetzt. <bit> ist von 0 bis 7, <value> 0 oder 1.
- **REG**: Register

GET Der GET MUSS den Decoder direkt befragen. Eine Antwort DARF NICHT anhand von servergespeicherten Informationen ermittelt werden. Der Einsatz von Wildcards ist nicht zulässig.

```
GET <bus> SM <decoderaddress> <type> <1 or more values>
```

Die Bedeutung der Parameter ist wie bei SET. Das Ergebnis ist ein `100 INFO`, das den betreffenden Wert widergibt. Ein GET hat die Fehlermeldung `416 ERROR no data` als Ergebnis, wenn es keine Möglichkeit gibt, den Decoder auszulesen.

VERIFY

```
VERIFY <bus> SM <decoderaddress> <type> <1 or more values>
```

Wie bei GET MUSS der Decoder direkt befragt werden.

Die Bedeutung der Parameter ist wie bei SET. Das Ergebnis ist entweder ein `200 OK`, wenn der Parameter im Decoder dem angegebenen entspricht oder die Fehlermeldung `412 ERROR wrong value`.

TERM

```
TERM <bus> SM
```

Beendet den mit INIT begonnenen Programmiermodus auf dem betreffenden Bus. Eventuell unvollständige Programmierzyklen sind geeignet zu beenden.

INFO

```
100 INFO <bus> SM <decoderaddress> <type> <1 or more values>
101 INFO <bus> SM <protocol>
102 INFO <bus> SM
```

Die Bedeutung der Parameter ist wie bei SET bzw. INIT.

Gerätegruppe LOCK: Schreibsperren Mit Locks kann ein Client ein Gerät exklusiv für die betreffende Session sperren. Dies gilt auch für physisch identische Geräte, die in unterschiedlichen Gerätegruppen ansprechbar sind: Ein LOCK auf einen Decoder in der Gerätegruppe GL verhindert auch Schreibzugriff auf den gleichen Decoder in der Geräte SM! Anderen Sessions, ggf. auch des gleichen Clients, wird beim schreibenden Zugriff die Fehlermeldung `414 ERROR device locked` gemeldet. Die Ausführung dieser Befehle ist dann nur dem den Lock haltenden Clientsession gestattet. Lesezugriffe werden weiterhin von allen Clients akzeptiert und funktionieren ohne Einschränkungen. Ein Lock auf ein Gerät der LOCK-Gerätegruppe, auch eines anderen Busses, ist nicht zulässig. Ebensowenig kann ein Bus als ganzes gesperrt werden.

Der Grundzustand ist nicht gelockt".

Für die Geräte der Gerätegruppe GL ist für alle Clients IMMER der Notstop zulässig. Ein Lock durch eine andere Session hat in diesem Fall keine Wirkung. Hierbei MUSS die Lok nur den Notstop ausführen, alle anderen Parameter, auch wenn sie angegeben sein sollten, MÜSSEN unverändert bleiben. Insbesondere bleibt der LOCK erhalten.

Ein Lock wird automatisch mit dem Ende der Clientverbindung und dem Ablaufen der Sperrzeit aufgehoben.

SET

```
SET <bus> LOCK  <devicegroup> <addr> <duration>
```

Setzt einen Lock auf das adressierte Gerät. Zu beachten ist, das die LOCK-Geräte keine eigene Adresse haben, sie werden ausschließlich über das von ihnen gesperrte Gerät identifiziert.

Andere Clients können noch lesend und nur in sicherheitsrelevanten Bereichen schreibend zugreifen. Der LOCK wird für <duration> Sekunden gehalten. Nach Ablauf dieser Zeitspanne wird der LOCK automatisch vom Server beendet. Die Zeit beginnt mit dem setzen des LOCK bzw. dem letzten SET Befehl. Eine Dauer von 0 bedeutet eine unbefristete Spanne. Ein wiederholtes Absetzen des SET Befehls startet den Zeitgeber jeweils neu.

TERM

```
TERM <bus> LOCK <devicegroup> <addr>
```

Ein Lock wird auf das Gerät <devicegroup> <addr> des Busses <bus> entfernt.

INFO

```
100 INFO <bus> LOCK <devicegroup> <addr> <Duration> <sessionid>
101 INFO <bus> LOCK <devicegroup> <addr> <duration>
102 INFO <bus> LOCK <devicegroup> <addr>
```

Die SessionID ist hierbei die eindeutige Kennung der Session, die den Lock hält. Die <Duration> ist die verbleibende Zeitdauer

GET

```
GET <bus> LOCK <devicegroup> <addr>
```

Diesr Befehl liefern den aktuellen Lockstatus eines Gerätes als `100 INFO`-Zeile. Ist das angegebene Gerät nicht gelockt, wird die SessionID 0 angegeben.

Gerätegruppe POWER: Energieversorgung Die Energieversorgung wird für jeden Bus getrennt geschaltet

Der Grundzustand ist OFF.

Wenn der Server von Veränderungen der Energieversorgung durch die Anlage selbst erfährt (z.b. Kurzschlußerkennung), so ist die gleichbedeutend mit entsprechenden einem Befehl und MUSS im Infomodus mitgeteilt werden.

INIT

```
INIT <bus> POWER
```

Initialisiert die Stromversorgung, sie wird jedoch noch nicht eingeschaltet!

GET

```
GET <bus> POWER
```

Liefert den aktuellen Zustand der Energieversorgung als `100 INFO`.

SET

```
SET <bus> POWER ON|OFF [freetext]
```

Aktiviert (ON) oder deaktiviert (OFF) die Energieversorgung auf dem betreffenden Bus. Der <freetext> ist eine optionale Ergänzung, die bis zu 100 Zeichen umfassen kann. Sie wird im `100 INFO` an andere Clients gemeldet. Ein SRCP Server wertet diesen Parameter nicht aus.

TERM

```
TERM <bus> POWER
```

Beendet die Stromversorgung und schaltet sie ab.

INFO

```
100 INFO <bus> POWER ON|OFF <freetext>
101 INFO <bus> POWER
102 INFO <bus> POWER
```

Informiert über den aktuellen Zustand der Energieversorgung und eventuell vorhandene Zusatzangaben.

Gerätegruppe TIME: Zeitnormal Die Modellzeit ist eine gegenüber der Normalzeit um einen konstanten Faktor verzerrte Zeit. Die Zeitverzerrung wird durch einen Multiplikator und einen Divisor gegenüber dieser Normalzeit angeben. Der Zeitgeber ist nur im Bus 0 zulässig, es gibt nur einen.

Im Grundzustand ist der Zeitgeber gestoppt.

Der Zeitgeber ist optionaler Bestandteil. Wird er nicht unterstützt, so MUSS eine Fehlermeldung `422 ERROR unsupported device group` bei allen Kommandos generiert werden.

Die Zeiterfassung besteht aus den Angaben: Tag, Stunde, Minute und Sekunde. Eine Minute besteht aus 60 Sekunden, eine Stunde aus 60 Minuten und ein Tag aus 24 Stunden. Die Tage werden fortlaufend gezählt, eine Unterteilung in größere Zeitintervalle (Kalenderfunktionen) ist Aufgabe des Clients.

Bei Erreichen einer vollen Modellminute wird im INFO Modus die aktuelle Modellzeit ausgegeben. Die Befehle GET und WAIT werten zusätzlich die Modellsekunden aus.

Die Modellzeit errechnet sich wie folgt:

```
(Delta) Modellzeit = (Delta) Realzeit * FX / FY
```

Beispiele:

- FX=10 FY=1 -> Jede Realminute werden 10 Modellminuten generiert (also alle 6 Sekunden eine).
- FX=1 FY=10 -> Alle 10 Realminuten läuft eine Modellminute ab.
- FX=1 FY=1 -> Jede Realminute läuft eine Modellminute ab

INIT

```
INIT 0 TIME <fx> <fy>
```

Initialisiert den Zeitgeber mit der angegebenen Verzerrung. Der Zeitgeber wird nicht automatisch gestartet. Die Parameter müssen positive, von Null verschiedene Zahlen sein. Ein Server KANN den Wertebereich auf einen eingeschränkten Bereich (z.B. 1..10) begrenzen. Diese Werte MÜSSEN in der Dokumentation angegeben werden.

SET

```
SET 0 TIME <JulDay> <Hour> <Minute> <Second>
```

Setzt den Zeitgeber auf den angegebenen Zeitpunkt und startet ihn. Bei einem laufenden Zeitgeber wird die neue Zeit mit dem Ablauf der laufenden Modellsekunde wirksam, so daß die neu beginnende Modellsekunde der neuen Zeit entspricht.

GET

```
GET 0 TIME
```

Liefert die aktuelle Modellzeit als `100 INFO`.

WAIT

```
WAIT 0 TIME <JulDay> <Hour> <Minute> <Second>
```

Wartet, bis die Modellzeit den angegebenen Zeitpunkt erreicht oder überschritten hat und liefert einen `100 INFO`-String mit der dann aktuellen Modellzeit.

Bei nicht laufendem Zeitgeber wird eine Fehlermeldung generiert: `416 ERROR no data`. Ist die aktuelle Modellzeit bereits später als die angegebene Zeit, ist die Bedingung ohne weitere Wartezeit erfüllt. Offensichtlich falsche Zeitangaben werden durch `412 ERROR wrong value` an den anfordernden Client gemeldet und ignoriert. Der WAIT MUSS immer die aktuell gültige Modellzeit auswerten, die ggf. durch SET verändert werden kann.

TERM

```
TERM 0 TIME
```

Stoppt und entfernt den Zeitgeber. Alle laufenden WAIT werden mit einem TIMEOUT beendet.

INFO

```
100 INFO 0 TIME <JulDay> <Hour> <Minute> <Second>
101 INFO 0 TIME <fx> <fy>
102 INFO 0 TIME
```

Liefert alle Parameter und aktuellen Werte des Zeitgebers.

SESSION Sessions sind aktive Verbindungen eines Clients mit einem Server. Ein Client KANN mehrere Sessions bei einem Server haben. Diese sind für den Server voneinander unabhängig. Eine Session wird anhand einer SessionID von anderen unterschieden. Diese SessionID wird während des Handshake vom Server erzeugt und ist für die Laufzeit des Servers eindeutig. Eine einmal vergebene SessionID kann von einem Server nicht ein zweites Mal vergeben werden.

Sessions sind nur im Bus 0 zulässig. Ein Grundzustand ist nicht definiert.

GET

```
GET 0 SESSION <SESSIONID>
```

Liefert Angaben zur Session als INFO.

TERM

```
TERM 0 SESSION [<sessionid>]
```

Beendet die aktuelle oder die angegebene SESSION und schließt die Socketverbindung. Als letzte Meldung vom Server an den Client erfolgt eine Quittung des Befehls `200 OK`.

INFO

```
100 INFO 0 SESSION <SESSIONID> [optionale weitere Parameter]
101 INFO 0 SESSION <SESSIONID> [optionale weitere Parameter]
102 INFO 0 SESSION <SESSIONID> [optionale weitere Parameter]
```

Die Angaben für die Session werden geliefert. Die weiteren Parameter können die Art der Session, IP Adresse des Clients, die Portnummer und weitere Angaben enthalten.

DESCRIPTION Die DESCRIPTION Gerätegruppe beschreibt andere Gerätegruppen und Busse. Sie informieren den Client über die grundsätzlichen Eigenschaften.

GET Dieser Befehl kommt in zwei Versionen vor: Mit kurzer Parameterliste und mit langer. Die kurze Parameterliste enthält nur die Busnummer. Mit dieser informiert der Server den Client über den betreffenden Bus. Die lange Parameterliste enthält Gerätergruppe und ggf Geräteadresse. Mit dieser wird das betreffende Gerät bzgl. seiner Initialisierung beschrieben.

```
GET <bus> DESCRIPTION
--> INFO <bus> DESCRIPTION <list of devicegroups>
```

Der INFO umfaßt in einer Zeile alle von dem betreffenden Bus unterstützten Gerätegruppen. Beispiel:

```
GET 0 DESCRIPTION
--> 100 INFO 0 DESCRIPTION SERVER SESSION TIME
GET 1 DESCRIPTION
--> 100 INFO 1 DESCRIPTION FB POWER
```

Der Bus 0 unterstützt die Gerätegruppen SERVER, SESSION und TIME, der Bus 1 nur Rückmelder (und natürlich POWER).

In der zweiten Variante wird die Adresse des betreffenden Gerätes angegeben und als Antwort die bei der Initialisierung definierten Parameter angegeben.

```
GET <bus> DESCRIPTION <devicegroup> [<address>]
```

Die Adresse DARF NUR bei Gerätegruppen ohne eine Adressierung weggelassen werden. Beispiel:

```
GET 1 DESCRIPTION GL 1
--> 101 INFO 1 GL 1 N 128 5
```

d.h. die Lok mit der Adresse 1 ist eine NMRA kompatibler Dekoder mit 128 realen Fahrstufen und 5 Funktionen.

SERVER Diese Gerätegruppe ist nur im Bus 0 definiert. Zulässig sind die Befehle TERM und RESET. Der Grundzustand ist RUNNING.

Bei Programmstart des Servers KÖNNEN Befehle intern bereits ausgeführt werden, ohne dass ein Client diese ausdrücklich anweisen muss. In jedem Fall müssen alle Änderungen gegenüber dem in dieser Norm festgelegten Grundzustandes im INFO-Modus bereitgestellt werden.

Das Programmende des Servers wird durch den Befehl `TERM 0 SERVER` bewirkt. Eine Quittung (`200 OK`) erfolgt auch in diesem Fall, allerdings vor der Ausführung. Anschließend wird die Verbindung vom Server beendet (dies gilt für alle angeschlossenen Clients. Im INFO-Modus befindliche SESSIONs werden mindestens eine Sekunde vor dem Programmende über das bevorstehende Programmende informiert.

TERM

```
TERM 0 SERVER
```

Beendet den laufenden Server und schließt alle Verbindungen. Die angeschlossenen Busse SOLLEN abgeschaltet und die aktiven Geräte in den Grundzustand versetzt werden. Für den anfordernden Client wird die Quittung `200 OK` gesendet. Sessions im INFO Modus werden über den laufenden TERM unterrichtet.

RESET

```
RESET 0 SERVER
```

Re-initialisiert den Server. Alle (aktiven) Geräte werden mittels RESET in den Grundzustand versetzt. Während eines RESET MÜSSEN alle Kommandos abgewiesen werden (`413 temporarily prohibited`). Bestehende Sessions, auch im Handshake, bleiben unberührt.

INFO

```
100 INFO 0 SERVER <Zustandsinfo>
```

Informiert über den aktuellen Zustand des Servers. Folgende Meldungen sind definiert:

- RUNNING: Server im Normalbetrieb
- RESETTING: Server führt einen Reset aus, nach Abschluß des RESET wird der Zustand RUNNING wieder eingenommen.
- TERMINATING: Server wird beendet.

4.3 Infomodus

Im Infomodus werden unidirektional alle Veränderungen aller Geräte vom Server an den angeschlossenen Client gesendet. Dem Client ist es VERBOTEN, Kommandos oder anderes an den Server zu senden. Der Server muß sicherstellen, das trotzdem gesendete Daten keine Auswirkungen haben; der Server KANN in diesem Fall die Verbindung seinerseits beenden.

Der Infomodus sendet alle Angaben so schnell wie möglich. Sie werden genauso wie die entsprechenden Angaben des Kommandomodus (siehe Antworten auf GET Befehle) immer mit dem Zeitstempel versehen. Es werden ausschließlich INFO Angaben gesendet. Insbesondere werden die Wirkungen von Kommandos erst dann gemeldet, wenn sie entweder vom Aktor (Dekoder) bestätigt wurden bzw. wenn sie erfolgreich an die Anlage übermittelt wurden (Wenn es keine Bestätigungen von der Anlage geben kann).

Es erfolgen keine Umwandlungen oder Interpretationen der Daten. Dies betrifft insb. die Geschwindigkeit von Lokomotiven, die von Clients zwar virtuell gesendet werden, jedoch vom Server auf die realen Gegebenheiten des betreffenden Decoders umgerechnet werden.

Beim Aktivieren des Infomodus in einer Session werden für alle Busse die DESCRIPTIONS und alle dem Server bekannten und aktiven Geräte mit ihrem aktuellem Zustand übermittelt. Anschließend werden alle Veränderungen gesendet. Wenn für einzelne Gerätegruppen ein eindeutiger Grundzustand existiert, so SOLLEN alle Geräte, die sich in diesem Grundzustand befinden nicht übermittelt werden. Optionale Gerätegruppen DÜRFEN NUR angegeben werden, wenn sie auch unterstützt werden.

Besonderheiten der Gerätegruppen bei Aktivieren des Infomodus:

FB Alle Geräte, die im Zustand 0 sind, werden nicht gemeldet.

GA Die einzelnen Ports der dem Server bekannten Geräte werden getrennt mit ihrer jeweils letzten Aktivität übermittelt. Wenn unterschiedliche Ports angesprochen wurden, so MUSS für jeden Port getrennt der INFO mit dem Zeitstempel der letzten Aktivierung gesendet werden. Dies SOLL in der korrekten zeitlichen Reihenfolge geschehen. Es MÜSSEN auch Informationen über Geräte übermittelt werden, die sich zwar gegenwärtig im Grundzustand befinden, die jedoch bereits mind. einmal aktiviert wurden und nicht mit TERM beendett wurden.

GL,SM Alle in Bewegung befindlichen oder mit einer aktivierten Funktion stehenden Loks werden gemeldet. Stillstehende Loks MÜSSEN gemeldet werden, wenn sie wenigsten einmal angesprochen und nicht mit TERM entfernt wurden. Ebenso werden Veränderungen durch INIT gemeldet. Bei SM werden alle Programmieraktionen gemeldet. Eine abgeschlossene Programmieraktion DARF bei neuen Verbindungen NICHT gesendet werden.

POWER Der Zustand der Energieversorgung MUSS angegeben werden.

TIME Die aktuelle Modellzeit MUSS übermittelt werden, sofern dieses Feature unterstützt wird und der Zeitgeber läuft.

SESSION Eine Identifizierung aller derzeit aktiven Clientsessions wird gesendet. Alle weiteren Informationen MÜSSEN im Kommandomodus ermittelt werden.

LOCK Alle derzeit gelockten Geräte werden gemeldet.

Ab der initialen Übermittlung aller Zustände MÜSSEN alle Veränderungen, auch solche in den Grundzustand, an der Anlage und dem Server übermittelt werden. Ausnahme ist die Gerätegruppe TIME: Sie meldet jeden Start einer neuen Modellminute, nicht jedoch jeder Modellsekunde.

Bei der Initialisierung und Beendigung von Bussen MÜSSEN die jeweils definierten `101 INFO`-Meldungen generiert werden. Bei der TERMinierung die `102 INFO`-Meldungen.

5 Security

Es werden keine sicherheitsrelevanten Maßnahmen vorgesehen.

6 Glossar

Die hier angeführten Begriffe bilden Schnittstellen zu anderen Dokumenten.

Julianisches Datum Das Julianische Datum ist eine einfache Tageszählung ohne eine weitere Strukturierung (etwa in Wochen/Monate/Jahre) vorzunehmen. Je nach Nullpunkt (Jahresanfang, fester Zeitpunkt in der Vergangenheit) existieren unterschiedliche Umrechnungsmöglichkeiten in andere Kalender (Basis: 1.1.4713 v.Chr. 12-00 GMT; siehe Collected Algorithms from CACM #199)

XRCP Eine Ergänzung des SRCP, um ein höheres Abstraktionsniveau zu schaffen. Neben der Anbindung der Clients an den Server dient es zur Kommunikation der Clients untereinander. Ein XRCP System ist als Middleware zwischen Clients und SRCP Servern gestaltet.