Spezifikation des Basic Serial Protocol (BSP)	
Version	5
Datum	09.04.18
Index	1 Übersicht 2 Verpflichtungen 2.1 Empfänger 2.2 Sender 3 Datenpakete 3.1 Small Data Packet 3.2 Medium Data Packet 3.3 Large Data Packet 3.4 Checksums 3.4.1 XOR Checksum 3.4.1 Cyclic Redundancy Check (CRC) 3.5 Paket Ketten 4 Das Antwortpaket 5 Ausnamesituationen 5.1 Unbekannter Packet Type 5.2 Reset Packet 5.3 Failed Checksum in einem AP 5.4 Wiederholtes Empfangen eines validen Pakets 6 Appendices A Packet Types B Packet ID C Cyclic Redundancy Check (CRC32) D XOR Checksum E Reset Acknowledge Sequence F Glossar

1 Übersicht

Das BSP dient zur Übertragung eines beliebig großen binären “Blobs” über eine etablierte Verbindung. Dabei ist das Ziel eine zuverlässige Datenübertragung zu gewärleisten.

Kommunikation erfolgt durch Austausch von Paketen zwischen zwei Klienten. Die zu übertragenden Daten werden von Klient #1 dem Sender-Program übergeben, welches diese in einem Datenpaket verpackt und verschickt. Das Empfänger-Program vom Klient #2 quittiert den Empfang des Datenpaketes mit dem verschicken eines Antwortpaketes an den Sender und übergibt die entpackten Daten an Klient #2.

2 Verpflichtungen

2.1 Empfänger

1. Der Empfänger ist verpflichtet jedes empfangene Datenpaket zu validieren und zu quittieren.
2. Der Empfänger muss sicherstellen, dass ein valides Datenpaket, das mehr als einmal empfangen wird, nur einmal an den Klienten übergeben wird. Trotzdem muss jedes eingegangene Datenpaket quittiert werden.

2.2 Sender

1. Der Sender ist verpflichtet darauf zu achten, dass der Empfänger zeitnah das verschickte Datenpaket quittiert und ggf. das Datenpaket erneut zu verschicken.
2. Der Sender ist verpflichtet ein Datenpaket, das nicht als OK quittiert wurde erneut zu verschicken.
3. Der Sender darf kein Datenpaket verschicken sofern nicht das vorherige Datenpaket quittiert wurde.

3 Datenpakete

Das Datenpaket besteht neben den zu übertragenden Daten (*raw data*) aus Metadaten, die dem Konrollfluß sowie Fehlererkennung dienen. Diese Metadaten werden als Header am Anfang eines Packetes verschickt und beeinhalten eine Paket Typkennung, eine Paket ID, die Länge der im Packet übermittelten Daten sowie Checksums zur Überprüfung der

Paketintegrität. Das Übertragen von Paketen mit variabler Länge ist unsicher, daher haben alle Pakete eine feste Länge und müssen ggf. mit “Füllbytes” aufgefüllt werden. Füllbytes können einen beliebigen Wert haben und werden vom Empfänger ignoriert. Um das Verhältnis von Daten zu Füllbytes zu maximieren werden drei verschiedene Datenpakete genutzt, abhängig von der Länge der *raw data*: Small Data Packet (SDP), Medium Data Packet (MDP) und Large Data Packet (LDP).

	SMD	**MDP**	**LDP**
Länge von *raw data*	1 - 16 Bytes	17 - 64 Bytes	65 - 512 Bytes
Länge von Header	6 Bytes	7 Bytes	10 Bytes
Füllbytes	0 - 15 Bytes	0 - 47 Bytes	0 - 447 Bytes
Gesamte Paketgröße	22 Bytes	71 Bytes	522 Bytes

3.1 Small Data Packet

	Größe	**Beschreibung**
Header	2 Bytes	Packet Type *Siehe Appendix A*
	1 Byte	Packet ID *Siehe Appendix B*
	1 Byte	Länge der *raw data*
	1 Byte	CRC8 Checksum
	1 Byte	XOR Checksum *Siehe Appendix D*
Payload	1 - 16 Bytes	*raw data*
	0 - 15 Bytes	Füllbytes (dürfen beliebige Werte haben)

3.2 Medium Data Packet

	Größe	Beschreibung
Header	2 Bytes	Packet Type *Siehe Appendix A*
	1 Byte	Packet ID *Siehe Appendix B*
	1 Byte	Länge der *raw data*
	2 Bytes	CRC16 Checksum
	1 Byte	XOR Checksum *Siehe Appendix D*
Payload	17 - 64 Bytes	*raw data*
	0 - 47 Bytes	Füllbytes

3.3 Large Data Packet

	Größe	Beschreibung
Header	2 Bytes	Packet Type *Siehe Appendix A*
	1 Byte	Packet ID *Siehe Appendix B*
	2 Bytes	Länge der *raw data*
	4 Bytes	CRC32 Checksum
	1 Byte	XOR Checksum *Siehe Appendix D*
Payload	65 - 512 Bytes	*raw data*
	0 - 447 Bytes	Füllbytes

3.4 Checksums

3.4.1 XOR Checksum

Es werden einfache XOR Checksums *Siehe Appendix D* zur Prüfung der Integrität der Headers verwendet.

3.4.2 Cyclic Redundancy Check (CRC)

Zur Überprüfung der Integrität der *raw data* übertragender Datenpakete werden Checksums nach ISO-3309 verwendet. *Siehe Appendix C* Dazu wird erst eine CRC32 Checksum des Datenpaketes berechnet. Die CRC16 Checksum wird durch das Verknüpfen beider Hälften der CRC32 Checksum berechnet. Die CRC8 Checksum wird analog dazu aus der CRC 16 Checksum berechnet.

3.5 Paket Ketten

Zur Übertragung größerer Datenmengen können Paketketten verwendet werden. Dabei wird eine Datengröße von 0 im Header eines Datenpaketes angegeben. Dies signalisiert, dass der Inhalt des Datenpaketes nur ein Teil der zu übertragenden *raw data* ist, der Empfänger reicht also die Daten nicht direkt weiter, sondern wartet auf das nächste Datenpaket. Paket Ketten können beliebig lang sein. Es gibt keine Einschränkungen bzgl. der Pakettypen innerhalb eine Paket Kette (zB eine Paket Kette von LDP, SDP, LDP ist legitim).

4 Das Antwortpaket

Jedes empfangene Paket wird vom Empfänger durch das senden eines Antwortpaketes (AP) quittiert. Ein AP ist exakt 4 Bytes groß und ist wie folgt aufgebaut:

Größe	Beschreibung
2 Byte	Packet Type *Siehe Appendix A*
1 Byte	Packet ID *Siehe Appendix B*
1 Byte	XOR Checksum *Siehe Appendix D*

Die Packet ID ist die ID des Datenpaketes auf das sich das AP bezieht.

5 Ausnahmesituationen

5.1 Unbekannter Packet Type

Wenn ein Paket mit einem Packet Type empfangen wird, der nicht in der Tabelle in Apendix A aufgeführt ist, ist diese unbekannte Type-ID zu analysieren. Weicht sie in Zwei oder weniger bits von einem der aufgeführten Werte ab ist anzunehmen, dass diese Type-ID gemeint war. In diesem Fall kann die empfangene Type-ID korrigiert werden und der Rest des Paketes normal eingelesen werden. Kann der ursprüngliche Packet Type nicht rekonstruiert werden, dann wird ein Reset Packet (RP) an den Sender geschickt. Daraufhin sind alle ankommenden Daten zu verwerfen bis die Reset Acknowledge Sequence (RAS) *Siehe Appendix E* empfangen wird.

5.2 Reset Packet

Das Empfangen eines RP signalisiert einen schwerwiegenden Datenverlust bei der Übertragung und bedeutet, dass keine Pakete bis zur Wiederherstellung der Kommunikation mehr verschickt werden dürfen. Wenn eine Quittierung für ein verschicktes Datenpaket noch aussteht ist dieses als verloren anzusehen und erneut zu senden. Wenn auf das letzte versandte AP kein Datenpaket gefolgt hat ist dieses AP ebenfalls erneut zu verschicken. Die Kommunikation wird wieder hergestellt durch das zweimalige Versenden der RAS.

5.3 Failed Checksum in einem AP

Sollte ein AP den Checksum Test nicht bestehen, dann gilt das noch nicht quittierte Datenpaket, das den Empfänger zum versenden des AP veranlasst hat, als nicht erfolgreich übertragen und muss noch einmal gesendet werden.

5.4 Wiederholtes Empfangen eines validen Paketes

Sollte ein Paket empfangen werden, was bereits vorher in einem validen Zustand empfangen wurde (zB durch eine Korruption eines AP), dann ist der Inhalt des Pakets zu ignorieren, aber trotzdem (im Falle eines Datenpakets) zu quittieren.

6 Appendices

A Packet Type

Ein Packet Type ist eine 2 Byte Sequenz die die Inhalt eines Pakets kategorisiert. Es bildet den Anfang von jedem Paket und sind im Big Endian Format. Es gibt 6 verschiedene Paket-Typen:

Wert	Paket	Erläuterung
81 A5	Small Data Packet	-/-
42 96	Medium Data Paket	-/-
24 C9	Large Data Paket	-/-
18 7C	Answer Packet (OK)	Datenpaket ist empfangen und valide.
36 13	Answer Packet (ERR)	Datenpaket ist beschädigt und muss noch einmal versandt werden.
70 B1	Reset Packet	Es gab einen schwerwiegenden Kommunikationsfehler und die beiden Klienten müssen sich erst synchronisieren bevor die Kommunikation wieder aufgenommen werden kann.

B Packet ID

Beim Senden eines Datenpaketes gibt der Sender diesem eine numerische ID. APs halten die ID des Paketes auf das sie sich beziehen. Die vergebenen IDs fangen bei 0 an und werden für jedes weitere **versandte** Paket um 1 erhöht. Da für das Packet ID nur ein Byte reserviert ist wird nach erreichen der ID 255 wieder bei 0 angefangen. Der Empfänger muss in der Lage sein, ein bereits korrekt Übertragenes Paket bei einer Neuübertragung zu erkennen und den Inhalt nicht weiter verarbeitet (auch dieses Paket muss quittiert werden!).

C Cyclic Redundancy Check (CRC32)

Die Integrität der übertragenden *raw data* wird durch einen CRC sichergestellt. Dabei wird vom Sender ein 32 bit Wert errechnet und im CP verschickt. Beim Empfangen des Datenpaketes errechnet der Empfänger ebenfalls einen Wert basierent auf der

empfangenen*raw data*. Diese beiden Werte sind identisch wenn das Paket korrekt übertragen wurde. Die Errechnung des CRC Wertes erfolgt nach ISO 3309. Dabei wird folgendes CRC Polynom genutzt:

$$f(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Der CRC Wert wird vor der Berechnung mit 1sen initialisiert und nach der Berechnung invertiert. Weder der Dateneingang von der Datenausgang des CRC Generators werden reflektiert.

D XOR Checksum

Die XOR Checksum beschreibt einen 1 Byte großen Wert der durch die Verknüpfung mit XOR aller anderen Bytes im Header entsteht. Dies hat zur Folge, dass wenn mal alle Bytes in einem Header (inklusive der XOR Checksum) durch XOR verknüpft der Wert 0x00 heraus kommt. Ist das Ergebnis dieser Berechnung *nicht* 0x00, dann wurde der Inhalt des Headers im Transport beschädigt.

E Reset Acknowledge Sequence

Die RAS ist die Folgende Byte-Sequenz:

```
91 0D 91 0D 91 0D 91 0D 91 0D 91 0D 91 0D 91 0D
```

Im Falle einer empfangenen RAS wird so lange weitergelesen, bis etwas anderes als eine Sequenz von `91 0D` gelesen wird.

F Glossar

Acronym	Bedeutung
BSP	Basic Serial Protocol
SDP	Small Data Packet
MDP	Medium Data Paket
LDP	Large Data Packet
CRC	Cyclic Redundancy Check
AP	Answer Packet
RP	Reset Packet
RAS	Reset Achnowledge Sequence