

# Datenprotokolle

für den direkt digitalisierenden Blackbox-Empfänger

## RSR200



Ausgabe: 0.40  
Erstellt: 25.11.2025  
Letzte Änderung: 06.02.2026

# Inhalt

1. Grundlagen.....	4
2. Schnittstellen.....	5
2.1 USB.....	5
2.1.1 Datenausrichtung bei Einkanalbetrieb 16 Bits.....	6
2.1.2 Datenausrichtung bei Zweikanalbetrieb 16 Bits.....	6
2.1.3 Datenausrichtung bei Einkanalbetrieb 24 Bits.....	6
2.2 LAN.....	7
2.2.1 TCP Datenübertragung.....	7
2.2.1.1 TCP-Blockstruktur bei Einkanalbetrieb 16 Bits.....	8
2.2.1.2 TCP-Blockstruktur bei Zweikanalbetrieb 16 Bits.....	9
2.2.1.3 TCP-Blockstruktur bei Einkanalbetrieb 24 Bits.....	10
2.2.2 UDP Datenübertragung.....	11
2.2.2.1 UDP-Blockstruktur bei Einkanalbetrieb 16 Bits.....	12
2.2.2.2 UDP-Blockstruktur bei Zweikanalbetrieb 16 Bits.....	13
2.2.2.3 UDP-Blockstruktur bei Einkanalbetrieb 24 Bits.....	14
3. Kommandos.....	15
3.1 Handling der im Datenstrom eingebetteten Kommandos.....	15
3.2 Kommandos RSR200 → PC.....	16
3.3 Kommandos PC → RSR200.....	18
3.4 Firmware Update.....	22
3.5 Kommando-Wiederholungen.....	24
4. Hinweise zum Betrieb des RSR200.....	25
4.1 Verwendung der USB-Schnittstelle.....	25
4.2 Verwendung der LAN-Schnittstelle mit TCP.....	26
4.3 Verwendung der LAN-Schnittstelle mit UDP.....	27
4.4 Antennensteuerung.....	28
4.5 Verwendung des GPS-Empfängers.....	28
4.6 Zweikanalbetrieb.....	29

## Historie

Anderung Nr.	Beschreibung	Firmware
01	Erstveröffentlichung	221
02	<ul style="list-style-type: none"><li>- Änderung Polarität Bit 2 Variable Nr. 5 "Schalter" des Kommandos "Variablen 16 Bit-Wert Einstellen"</li><li>- Erhöhung der Auflösung des Wertes für die ADC-Taktfrequenz auf 0,1 MHz im Kommando "ADC-Takt Einstellen"</li></ul>	222
03	<ul style="list-style-type: none"><li>- Erweitertes Blockschaltbild der Signalverarbeitung im FPGA</li></ul>	222
04	<ul style="list-style-type: none"><li>- Weitere Hinweise zu Betriebsart "Separate" (Kommando "Datenübertragung Einstellen → DSP-Modus" und Kommando „ADC-Takt Einstellen“</li><li>- Erläuterungen zur den Kommando-Nummern Abschnitt ("Kommandos")</li><li>- Neuer Abschnitt "Zweikanalbetrieb"</li></ul>	223

# 1. Grundlagen

Der RSR200 besitzt eine USB- und eine Netzwerk- (LAN-) Schnittstelle zur Verbindung mit einem oder zwei PC. Daten werden über diese Schnittstellen in beide Richtungen übertragen:

RSR200 → PC: IQ-Daten aus den ADC und der nachfolgenden Signalverarbeitung, Messwerte, Bestätigungen für ausgeführte Steuer-Kommandos.

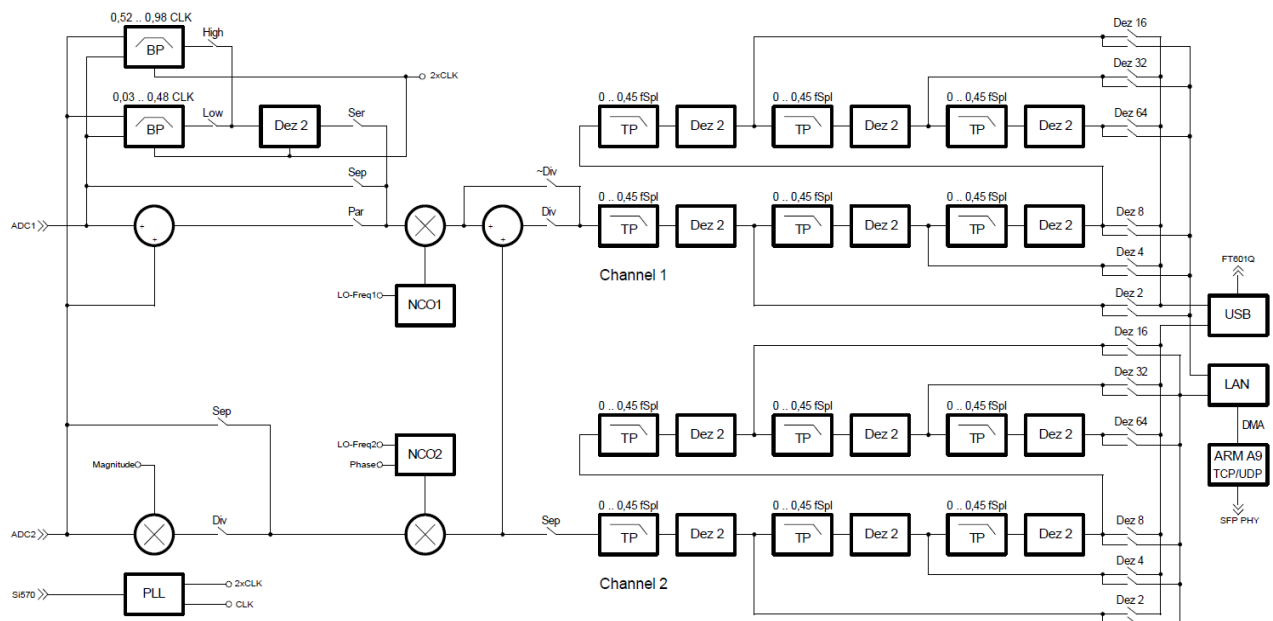
PC → RSR200: Steuer-Kommandos, Firmware-Dateien.

Die Dateninhalte sind für beide Schnittstellen weitgehend gleich. Die Formatierung der Daten (Zahl der übertragenen Bytes, Anordnung der Daten in den übertragenen Bytes) variiert zwischen den Schnittstellen und den verwendeten Protokollen.

Die Datenübertragung vom RSR200 zum PC ist auf höchstmögliche Geschwindigkeit (maximale Bandbreite der IQ-Daten) optimiert. Die Daten werden in Form von Paketen und Blöcken (mehrere zusammengehörende Pakete) übertragen. In den Paketen sind Nutz- (IQ-) und Steuerungsdaten (Kommandos) enthalten. Pakete bzw. Blöcke werden regelmäßig vom RSR200 an den PC gesendet („Streaming“). Die Paket- / Block-Rate hängt von der Zahl der im RSR200 erzeugten Nutzdaten (Sampling-Rate) und weiteren Einstellungen (z. B. Bitbreite der Daten, Zahl der Kanäle) ab. Der PC muss in der Lage sein, alle Pakete / Blöcke im Zeitraster der Sampling-Rate empfangen zu können, sonst treten Datenverluste auf.

Zur Beschreibung der Funktionalität des RSR200 ist dessen Handbuch und darin insbesondere das Blockschaltbild zu verwenden. Aus der Erläuterung der Bedienelemente sind Hinweise auf sinnvolle Einstellmöglichkeiten und die Befehlsstruktur in den Datenpaketen ersichtlich.

Werte in den Datenpaketen werden dezimal (einfache Zahlenwerte) oder hexadezimal in C-Notierung (vorangestelltes 0x) dargestellt.



Hardware der digitalen Signalverarbeitung im FPGA des RSR200.

## 2. Schnittstellen

### 2.1 USB

Die USB-Schnittstelle arbeitet nach dem Standard USB 3.0 (5 GBit). Im RSR200 wird der Schaltkreis FT601Q der Firma FTDI verwendet. Die Software des PCs sollte die vom Hersteller für diesen Schaltkreis erstellte API verwenden. Sie enthält eine Bibliothek mit Routinen zur Verbindung und Datenübertragung eines USB 3.0 konformen Ports am PC mit dem RSR200.

Der RSR200 betreibt den FT601Q mit folgenden Einstellungen:

- Ein Bulk-Transmit-Endpoint (0x82) mit 4096 Byte Doppel-Pufferspeicher
- Ein Bulk-Receive-Endpoint (0x02) mit 512 Byte Puffer + FPGA-interner 264 Byte Puffer.

Nach dem Einschalten beginnt der RSR200 sofort mit der Verarbeitung von ADC-Daten und der Füllung des Transmit-Speichers. Sobald ein Puffer voll ist, wird der Endpoint als „Ready“ gekennzeichnet und die entsprechende Routine im PC (einer der Read-Befehle der FT601Q API) kann den Puffer lesen. Dies muss in einer kürzeren Zeit erfolgen, als sie der 2. Pufferspeicher im RSR200 zum Vollwerden benötigt. Dann ist eine fortlaufende Übertragung (Streaming) ohne fehlende Daten möglich (jeweils 1 Puffer wird gefüllt, solange der andere vom PC gelesen wird). Die Zeit zum Füllen eines Puffers errechnet sich aus der eingestellten Samplingrate, der Bitbreite und der Kanalzahl der Nutzdaten.

*Hinweis:* Das Lesen eines Pakets, die Datenverarbeitung und die erneute Anforderung von Daten benötigt im PC Rechenzeit. Die FTDI-API enthält Möglichkeiten, das Lesen mehrerer Pakete mit minimalem Overhead auszuführen. Dazu kann die Pipe zum Lesen von Endpoint 0x82 auf „Streaming“ gesetzt und asynchrones Lesen („Overlapped“ Mechanismus) verwendet werden. Der Lesebefehl sollte mit möglichst vielen Einzelpuffern von 4096 Byte Größe beauftragt werden. Für geringsten Overhead sollte das Warten auf die Füllung der Puffer nicht unterbrochen werden (Parameter Overlapped.Result = True). Das führt allerdings zum Blockieren des Programms, falls Fehler auftreten!

Ein Paket = ein Block (ein Pufferinhalt) der USB-Schnittstelle hat folgende Struktur:

Byte-Nr.	Wert	Bedeutung
0	LSB Paketzähler	Fortlaufender Paket-Zähler 32 Bit
:	:	
3	MSB Paketzähler	
4	1. Datenbyte	4080 Bytes IQ-Daten
:	:	
4083	4080. Datenbyte	
4084	Signed Byte	Temperatur in °C
4085	LSB GPS	Aktueller Wert der Frequenzkorrektur in Hz als vorzeichenbehafteter 14 Bit-Wert, OV1, OV2
4086	MSB GPS, OV	
4087	Kommando-Nr.	Nummer des aktuell übertragenen Kommandos
4088	LSB Kommando	8 Bytes aktuelles Kommando
:	:	
4095	MSB Kommando	

Die Daten sind grundsätzlich 32 Bit Wort-orientiert. Die Ausrichtung ist „little endian“, niederwertige Bytes werden zuerst übertragen bzw. liegen auf niedrigeren Adressen. Paketgröße = Blockgröße = 4096 Bytes = 1024 Words.

Das erste Datenwort beinhaltet einen Paketzähler, der in jedem folgendem Paket um 1 erhöht wird. Es folgen 1020 Worte = 4080 Datenbytes mit IQ-Daten (siehe unten). Das Ende des Pakets enthält Messwerte und das aktuelle Kommando vom RSR200 zum PC (siehe Abschnitt Kommandos).

In jedem Paket sind 4080 Bytes IQ-Daten enthalten. Die darin enthaltene Anzahl von Abtastwerten (Samples) schwankt abhängig von den Einstellungen. Die Daten sind je nach eingestellter Auflösung und Kanalzahl wie folgt organisiert.

## 2.1.1 Datenausrichtung bei Einkanalbetrieb 16 Bits

### IQ-Daten 1-Kanal 16 Bit: 1020 Samples

Byte-Nr.	Wert	Bedeutung
0	LSB I-Daten	16 Bit Inphase-Daten 1. Sample im Paket
1	MSB I-Daten	
2	LSB Q-Daten	16 Bit Quadratur-Daten 1. Sample im Paket
3	MSB Q-Daten	
:	:	:
4076	LSB I-Daten	16 Bit Inphase-Daten 1020. Sample im Paket
4077	MSB I-Daten	
4078	LSB Q-Daten	16 Bit Quadratur-Daten 1020. Sample im Paket
4079	MSB Q-Daten	

## 2.1.2 Datenausrichtung bei Zweikanalbetrieb 16 Bits

### IQ-Daten 2-Kanal 16 Bit: 510 Samples

Byte-Nr.	Wert	Bedeutung
0	LSB I-Daten	16 Bit Inphase-Daten 1. Sample Kanal 1
1	MSB I-Daten	
2	LSB Q-Daten	16 Bit Quadratur-Daten 1. Sample Kanal 1
3	MSB Q-Daten	
4	LSB I-Daten	16 Bit Inphase-Daten 1. Sample Kanal 2
5	MSB I-Daten	
6	LSB Q-Daten	16 Bit Quadratur-Daten 1. Sample Kanal 2
7	MSB Q-Daten	
:	:	:
4072	LSB I-Daten	16 Bit Inphase-Daten 510. Sample Kanal 1
4073	MSB I-Daten	
4074	LSB Q-Daten	16 Bit Quadratur-Daten 510. Sample Kanal 1
4075	MSB Q-Daten	
4076	LSB I-Daten	16 Bit Inphase-Daten 510. Sample Kanal 2
4077	MSB I-Daten	
4078	LSB Q-Daten	16 Bit Quadratur-Daten 510. Sample Kanal 2
4079	MSB Q-Daten	

## 2.1.3 Datenausrichtung bei Einkanalbetrieb 24 Bits

### IQ-Daten 1-Kanal 24 Bit: 680 Samples

Byte-Nr.	Wert	Bedeutung
0	LSB I-Daten	24 Bit Inphase-Daten 1. Sample im Paket
1	:	
2	MSB I-Daten	
3	LSB Q-Daten	24 Bit Quadratur-Daten 1. Sample im Paket
4	:	
5	MSB Q-Daten	
:	:	:
4074	LSB I-Daten	24 Bit Inphase-Daten 680. Sample im Paket
4075	:	
4076	MSB I-Daten	
4077	LSB Q-Daten	24 Bit Quadratur-Daten 680. Sample im Paket
4078	:	
4079	MSB Q-Daten	

## 2.2 LAN

Die LAN-Schnittstelle arbeitet nach Standard IEEE 802.3 (Ethernet) mit 1 GBit Datenrate. Die Datenübertragung erfolgt mit den Standard-Protokollen TCP und UDP. Der RSR200 richtet nach dem Einschalten einen TCP-Server ein und wartet auf eingehende Verbindungen. Der Server arbeitet auf Port 55557. Die IP-Adresse ist nichtflüchtig im Gerät gespeichert. Sie lautet standardmäßig 191.168.1.10 und kann per Steuer-Kommando auf beliebige Werte gesetzt werden. Für ca. 5 s nach dem Einschalten betreibt der RSR200 einen DHCP-Client. Während dieser Zeit kann die Adresse dynamisch (für die Einschaltdauer des RSR200) per DHCP-Dienst geändert werden.

Der TCP-Server sendet zunächst nur Bestätigungen (kurze TCP-Pakete ohne IQ-Daten) für eingehende Kommandos (siehe Abschnitt Kommandos). Per Kommando kann er auf einen Streaming-Betrieb umgestellt werden. Danach sendet er regelmäßig größere Blöcke von Nutz- und Steuerdaten (so, wie Nutzdaten verfügbar sind und damit die Blöcke gefüllt werden). Per Kommando kann wieder zurück in den Paket-Modus geschaltet werden.

Zusätzlich zum TCP-Server kann per Kommando ein UDP-Dienst eingerichtet werden. Der UDP-Dienst arbeitet an Port 55558 und empfängt / sendet immer Einzelpakete. Im Stream-Modus werden soviel logisch zusammen gehörende Einzel-Pakete gesendet, wie jeweils zur Übertragung eines kompletten LAN-Datenblocks (IQ- und Kommandodaten) notwendig.

*Hinweis:* TCP- und UDP-Transceiver sind im RSR200 ohne Betriebssystem (minimaler Overhead) direkt auf dem TCP-IP Stack „lwip220“ aufgebaut. Sie erreichen damit nahezu die maximal mögliche Netto-Datenrate von knapp 1 GBit. Der PC muss in der Lage sein, diese Datenrate zu empfangen, sonst gehen Daten verloren. Die Implementierung der zum RSR200 gehörenden „ExtIO-DLL“ arbeitet mit asynchronen TCP- bzw. UDP-Sockets aus der Microsoft WINSOCK2 API und erreicht damit bei unbelastetem PC (keine anderen Programme und keine weitere Signalverarbeitung) ca. 900 MBit Datenrate.

Die LAN-Schnittstelle arbeitet grundsätzlich mit Blöcken gleicher Anzahl enthaltener Samples (130560) und damit schwankender Blocklänge je nach eingestellter Auflösung und Kanalzahl (siehe unten). Daten sind grundlegend Byte-orientiert, Werte größer 1 Byte sind little-endian orientiert. Die IQ-Samples sind ab Anfang eines Blocks vorhanden. Darauf folgt ein fortlaufender 32 Bit Blockzähler (um 1 erhöht in jedem folgenden Block). Der Blockzähler ist in den nächsten 4 Bytes noch einmal, allerdings invertiert (Einer-Komplement) enthalten. Es folgen 8 Byte für Synchronisationszwecke (immer gleich in jedem Block). Das Ende des Blocks enthält Kommandos vom RSR200 zum PC (siehe Abschnitt Kommandos).

Zur Identifizierung der Block- (und bei UDP auch der Paket-) Grenzen in einem kontinuierlichen Datenstrom (Empfang beliebiger Zahl von Bytes aus den Sockets) können die Block- bzw. Paketzähler und die Synchronisations-Bytes benutzt werden. Mit Hilfe der Zähler und der bekannten festen Länge (je nach Betriebsart) der Blöcke ist auch eine Überprüfung auf fehlende Daten und damit auf die Qualität der Übertragung (Störungen, Unterbrechungen, oder Überschreiten der möglichen Übertragungsrate) möglich. Eine Fehlerüberprüfung der Daten und eventuelle Korrektur (Checksummen, FEC usw.) ist auf höherer Ebene nicht implementiert.

### 2.2.1 TCP Datenübertragung

TCP erlaubt das Senden von Datenblöcken (nahezu) beliebiger Größe. Der TCP-Stack übernimmt dabei das komplette Handling der Daten wie Paket-Generierung, Aufteilung größerer Datenmengen auf einzelne TCP-Pakete, Fehlererkennung / -korrektur usw. Beim RSR200 sind deshalb relativ große Datenblöcke als kleinste Einheit definiert (keine Einzelpakete). Dies erfordert relativ wenig Software-Overhead.

TCP arbeitet mit Prüfung der Daten und Acknowledgement, wodurch relativ sichere Datenübertragungen möglich sind (automatische Wiederholungen bei Fehlern). Das erfordert jedoch zusätzlichen Overhead und vermindert die Datenrate. Ganz besonders im Fall von Fehlern wird der Datenstrom so lange unterbrochen, bis das fehlerhafte Paket erneut und korrekt übertragen wurde. Dies behindert ein kontinuierliches Streaming von Echtzeitdaten (Datenüberlauf mit Verlusten im Sender).

*Hinweis:* Es gibt viele verschiedene „Stellschrauben“ für das TCP-Protokoll, mit denen das Acknowledgement-System getrimmt werden kann (z. B. Puffergrößen, Wartezeiten, Zahl der Pakete, die gleichzeitig bestätigt werden u.v.m.). Ein Großteil davon arbeitet automatisch innerhalb des TCP-Stacks bzw. des PC-Betriebssystems. Der Programmierer ist angehalten, optimale Werte zum Erreichen hoher Datenraten für die Stack-Einstellungen zu programmieren (soweit zugänglich).

**TCP-Block 1-Kanal 16 Bit: 522704 Bytes**

Byte-Nr.	Wert	Bedeutung
0	LSB I-Daten	16 Bit Inphase-Daten 1. Sample im Block
1	MSB I-Daten	
2	LSB Q-Daten	16 Bit Quadratur-Daten 1. Sample im Block
3	MSB Q-Daten	
:	:	:
522236	LSB I-Daten	16 Bit Inphase-Daten 130560. Sample im Block
522237	MSB I-Daten	
522238	LSB Q-Daten	16 Bit Quadratur-Daten 130560. Sample im Block
522239	MSB Q-Daten	
522240	LSB Blockzähler	Fortlaufender Block-Zähler 32 Bit
:	:	
522243	MSB Blockzähler	
522244	LSB invertierter Blockzähler	Fortlaufender Block-Zähler 32 Bit, invers
:	:	
522247	MSB invertierter Blockzähler	
522248	0x78	1. Synchronisations-Wort
522249	0x56	
522250	0x34	
522251	0x12	
522252	0xF0	2. Synchronisations-Wort
522253	0xDE	
522254	0xBC	
522255	0x9A	
522256	Signed Byte	Temperatur in °C
522257	LSB GPS	Aktueller Wert der Frequenzkorrektur in Hz als vorzeichenbehafteter 14 Bit-Wert, OV1, OV2
522258	MSB GPS, OV	
522259	Kommando-Nr.	Nummer des aktuell übertragenen Kommandoblocks
522260	LSB Kommando-Anzahl	32 Bit Zahl der folgenden Kommandos
:	:	
522263	MSB Kommando-Anzahl	
522264	Kommando	1. Byte (Befehl) 1. Kommando
:	:	:
522703	Datenbyte	Letztmögliches Byte für letztes Kommando



**TCP-Block 2-Kanal 16 Bit: 1045408 Bytes**

Byte-Nr.	Wert	Bedeutung
0	LSB I-Daten	16 Bit Inphase-Daten 1. Sample Kanal 1
1	MSB I-Daten	
2	LSB Q-Daten	16 Bit Quadratur-Daten 1. Sample Kanal 1
3	MSB Q-Daten	
4	LSB I-Daten	16 Bit Inphase-Daten 1. Sample Kanal 2
5	MSB I-Daten	
6	LSB Q-Daten	16 Bit Quadratur-Daten 1. Sample Kanal 2
7	MSB Q-Daten	
:	:	:
1044472	LSB I-Daten	16 Bit Inphase-Daten 130560. Sample Kanal 1
1044473	MSB I-Daten	
1044474	LSB Q-Daten	16 Bit Quadratur-Daten 130560. Sample Kanal 2
1044475	MSB Q-Daten	
1044476	LSB I-Daten	16 Bit Inphase-Daten 130560. Sample Kanal 2
1044477	MSB I-Daten	
1044478	LSB Q-Daten	16 Bit Quadratur-Daten 130560. Sample Kanal 2
1044479	MSB Q-Daten	
1044480	LSB Blockzähler	Fortlaufender Block-Zähler 32 Bit
:	:	
1044483	MSB Blockzähler	
1044484	LSB invertierter Blockzähler	Fortlaufender Block-Zähler 32 Bit, invers
:	:	
1044487	MSB invertierter Blockzähler	
1044488	0x78	1. Synchronisations-Wort
1044489	0x56	
1044490	0x34	
1044491	0x12	
1044492	0xF0	2. Synchronisations-Wort
1044493	0xDE	
1044494	0xBC	
1044495	0x9A	
1044496	Signed Byte	Temperatur in °C
1044497	LSB GPS	Aktueller Wert der Frequenzkorrektur in Hz als
1044498	MSB GPS, OV	vorzeichenbehafteter 14 Bit-Wert, OV1, OV2
1044499	Kommando-Nr.	Nummer des aktuell übertragenen Kommandoblocks
1044500	LSB Kommando-Anzahl	32 Bit Zahl der folgenden Kommandos
:	:	
1044503	MSB Kommando-Anzahl	
1044504	Kommando	1. Byte (Befehl) 1. Kommando
:	:	:
1045407	Datenbyte	Letztmögliches Byte für letztes Kommando

**TCP-Block 1-Kanal 24 Bit: 784784 Bytes**

Byte-Nr.	Wert	Bedeutung
0	LSB I-Daten	24 Bit Inphase-Daten 1. Sample im Block
1	:	
2	MSB I-Daten	
3	LSB Q-Daten	24 Bit Quadratur-Daten 1. Sample im Block
4	:	
5	MSB Q-Daten	
:	:	:
783354	LSB I-Daten	24 Bit Inphase-Daten 130560. Sample im Block
783355	:	
783356	MSB I-Daten	
783357	LSB Q-Daten	24 Bit Quadratur-Daten 130560. Sample im Block
783358	:	
783359	MSB Q-Daten	
783360	LSB Blockzähler	Fortlaufender Block-Zähler 32 Bit
:	:	
783363	MSB Blockzähler	
783364	LSB invertierter Blockzähler	Fortlaufender Block-Zähler 32 Bit, invers
:	:	
783367	MSB invertierter Blockzähler	
783368	0x78	1. Synchronisations-Wort
783369	0x56	
783370	0x34	
783371	0x12	
783372	0xF0	2. Synchronisations-Wort
783373	0xDE	
783374	0xBC	
783375	0x9A	
783376	Signed Byte	
783377	LSB GPS	Aktueller Wert der Frequenzkorrektur in Hz als vorzeichenbehafteter 14 Bit-Wert, OV1, OV2
783378	MSB GPS, OV	
783379	Kommando-Nr.	Nummer des aktuell übertragenen Kommandos
783380	LSB Kommando-Anzahl	32 Bit Zahl der folgenden Kommandos
:	:	
783383	MSB Kommando-Anzahl	
783384	Kommando	1. Byte (Befehl) 1. Kommando
:	:	:
784783	Datenbyte	Letztmögliches Byte für letztes Kommando

## 2.2.2 UDP Datenübertragung

UDP erlaubt je nach Implementierung des Stacks das Senden von Datenblöcken (nahezu) beliebiger Größe. Jedoch ist der Aufwand zur Fragmentierung eines Blocks in UDP-Pakete beim verwendeten lwip220-Stack erheblich und verringert die mögliche Datenrate. Im RSR200 erfolgt deshalb eine Aufteilung des LAN-Datenblocks durch die RSR200-Firmware in einzelne UDP-Pakete mit einer Länge, die gerade noch keine weitere Fragmentierung benötigen (1458 Bytes).

UDP-Pakete werden nicht auf Verlust oder enthaltene Übertragungsfehler geprüft und deshalb bei Fehlern nicht automatisch wiederholt. Es werden auch keine Acknowledgements vom Empfänger zurück an den Sender geschickt. Der Empfänger muss nicht auf diese Bestätigungen warten. Deshalb ist mit UDP ein kontinuierliches Streaming ohne Datenstopp bei Fehlern möglich. Treten Fehler auf, sind die Daten der betroffenen Pakete allerdings beschädigt oder unwiederbringlich verloren. Es wird deshalb empfohlen, UDP nur für die IQ-Daten zu verwenden und die mögliche Übertragung von Steuer-Kommandos nicht zu benutzen.

*Hinweis:* Der RSR200 sendet einzelne UDP-Pakete mit Kommandos nur für die Anforderung der Versionsnummer bei nicht eingeschaltetem Streaming. Im Streaming-Betrieb werden jedoch auch Messwerte und Kommando-Bestätigungen im LAN-Block gesendet.

Je nach Blocklänge wird eine gewisse Anzahl einzelner UDP-Pakete gesendet, die logisch zusammenhängen (ein „LAN-Block“). Die einzelnen Pakete müssen nach dem Empfang entsprechend ihrer Paketnummern wieder zu einem Block zusammengesetzt werden. Jedes Paket hat folgenden Aufbau:

### UDP Paket

Byte-Nr.	Wert	Bedeutung
0	LSB Paketnummer	16 Bit Nummer des Pakets im Block (ab 0)
1	MSB Paketnummer	
2	Daten	1. Datenbyte im Paket (fortlaufend ab Byte 0 des Blocks)
:	:	:
1457	Daten	Letztes Datenbyte im Paket

Das Paket beginnt mit einer 16 Bit Paketnummer (Position des Pakets im Block, mit 0 am Blockanfang beginnend), gefolgt von 1456 Byte der fortlaufenden Daten des Blocks. Dabei gibt es keinen Zusammenhang der Bytes mit Grenzen von Samples. Erst im wieder zusammengesetzten Block entsprechen bestimmte Byte-Nummern bestimmten Anfängen und Enden von Samples (siehe TCP-Block).

Bis auf das letzte Paket enthalten alle Pakete IQ-Daten. Das letzte Paket enthält die letzten IQ-Daten bis zum 130560. Sample des Blocks und im noch übrig bleibenden Teil die Steuer-Kommandos des Blocks.

**UDP-Block 1-Kanal 16 Bit: 359 Pakete = 523422 Bytes**

Byte-Nr.	Wert	Bedeutung
0	1. Byte des 1. Pakets	Anfang 1. UDP-Paket
:	:	:
1457	1458. Byte des 1. Pakets	Ende 1. UDP-Paket
:	:	:
521964	1. Byte des letzten Pakets	Anfang 359. UDP-Paket
:	:	:
523421	1458. Byte des letzten Pakets	Ende 359. UDP-Paket

**359. Paket im Block 1-Kanal 16 Bit**

Byte-Nr.	Wert	Bedeutung
0	0x66	16 Bit Nummer des Pakets: 358
1	0x01	
2	Daten	1. Datenbyte im Paket
:	:	:
993	Daten	Letztes Datenbyte im Paket und Block
994	LSB Blockzähler	Fortlaufender Block-Zähler 32 Bit
:	:	
997	MSB Blockzähler	
998	LSB invertierter Blockzähler	Fortlaufender Block-Zähler 32 Bit, invers
:	:	
1001	MSB invertierter Blockzähler	
1002	0x78	1. Synchronisations-Wort
1003	0x56	
1004	0x34	
1005	0x12	
1006	0xF0	2. Synchronisations-Wort
1007	0xDE	
1008	0xBC	
1009	0x9A	
1010	Signed Byte	Temperatur in °C
1011	LSB GPS	Aktueller Wert der Frequenzkorrektur in Hz als
1012	MSB GPS, OV	vorzeichenbehafteter 14 Bit-Wert, OV1, OV2
1013	Kommando-Nr.	Nummer des aktuell übertragenen Kommandoblocks
1014	LSB Kommando-Anzahl	32 Bit Zahl der folgenden Kommandos
:	:	
1017	MSB Kommando-Anzahl	
1018	Kommando	1. Byte (Befehl) 1. Kommando
:	:	:
1457	Datenbyte	Letztmögliches Byte für letztes Kommando

**UDP-Block 2-Kanal 16 Bit: 718 Pakete = 1046844 Bytes**

Byte-Nr.	Wert	Bedeutung
0	1. Byte des 1. Pakets	Anfang 1. UDP-Paket
:	:	:
1457	1458. Byte des 1. Pakets	Ende 1. UDP-Paket
:	:	:
1045386	1. Byte des letzten Pakets	Anfang 718. UDP-Paket
:	:	:
1046843	1458. Byte des letzten Pakets	Ende 718. UDP-Paket

**718. Paket im Block 2-Kanal 16 Bit**

Byte-Nr.	Wert	Bedeutung
0	0xCD	16 Bit Nummer des Pakets: 717
1	0x02	
2	Daten	1. Datenbyte im Paket
:	:	:
529	Daten	Letztes Datenbyte im Paket und Block
530	LSB Blockzähler	Fortlaufender Block-Zähler 32 Bit
:	:	
533	MSB Blockzähler	
534	LSB invertierter Blockzähler	Fortlaufender Block-Zähler 32 Bit, invers
:	:	
537	MSB invertierter Blockzähler	
538	0x78	1. Synchronisations-Wort
539	0x56	
540	0x34	
541	0x12	
542	0xF0	2. Synchronisations-Wort
543	0xDE	
544	0xBC	
545	0x9A	
546	Signed Byte	Temperatur in °C
547	LSB GPS	Aktueller Wert der Frequenzkorrektur in Hz als
548	MSB GPS, OV	vorzeichenbehafteter 14 Bit-Wert, OV1, OV2
549	Kommando-Nr.	Nummer des aktuell übertragenen Kommandoblocks
550	LSB Kommando-Anzahl	32 Bit Zahl der folgenden Kommandos
:	:	
553	MSB Kommando-Anzahl	
554	Kommando	1. Byte (Befehl) 1. Kommando
:	:	:
1457	Datenbyte	Letztmögliches Byte für letztes Kommando

**UDP-Block 1-Kanal 24 Bit: 539 Pakete = 785862 Bytes**

Byte-Nr.	Wert	Bedeutung
0	1. Byte des 1. Pakets	Anfang 1. UDP-Paket
:	:	:
1457	1458. Byte des 1. Pakets	Ende 1. UDP-Paket
:	:	:
784404	1. Byte des letzten Pakets	Anfang 539. UDP-Paket
:	:	:
785861	1458. Byte des letzten Pakets	Ende 539. UDP-Paket

**539. Paket im Block 1-Kanal 24 Bit**

Byte-Nr.	Wert	Bedeutung
0	0x1A	16 Bit Nummer des Pakets: 538
1	0x02	
2	Daten	1. Datenbyte im Paket
:	:	:
33	Daten	Letztes Datenbyte im Paket und Block
34	LSB Blockzähler	Fortlaufender Block-Zähler 32 Bit
:	:	
37	MSB Blockzähler	
38	LSB invertierter Blockzähler	Fortlaufender Block-Zähler 32 Bit, invers
:	:	
41	MSB invertierter Blockzähler	
42	0x78	1. Synchronisations-Wort
43	0x56	
44	0x34	
45	0x12	
46	0xF0	2. Synchronisations-Wort
47	0xDE	
48	0xBC	
49	0x9A	
50	Signed Byte	Temperatur in °C
51	LSB GPS	Aktueller Wert der Frequenzkorrektur in Hz als
52	MSB GPS, OV	vorzeichenbehafteter 14 Bit-Wert, OV1, OV2
53	Kommando-Nr.	Nummer des aktuell übertragenen Kommandoblocks
54	LSB Kommando-Anzahl	32 Bit Zahl der folgenden Kommandos
:	:	
57	MSB Kommando-Anzahl	
58	Kommando	1. Byte (Befehl) 1. Kommando
:	:	:
1457	Datenbyte	Letztmögliches Byte für letztes Kommando

### 3. Kommandos

Zur Steuerung des RSR200 ist es notwendig, außer IQ-Daten auch Kommando-Daten zu übertragen. Dazu existieren im RSR200 folgende Möglichkeiten:

- Senden von Daten vom RSR200 an den PC mittels einzelner TCP- oder UDP-Pakete bei ausgeschaltetem LAN-Streaming.
- Senden von in den Datenblöcken eingebetteten Kommando-Daten im Streaming-Betrieb (bei USB immer) vom RSR200 an den PC.
- Senden von einzelnen USB-, TCP- oder UDP-Paketen vom PC zum RSR200.

Das Senden einzelner Pakete an den RSR200 ist jederzeit möglich. Im PC-Programm muss dazu eine entsprechende Verbindung aufgebaut und das Kommando-Paket darüber gesendet werden. Bei USB sind die entsprechenden Befehle der FTDI-API benutzbar. Die Daten müssen an Endpoint 0x02 gesendet werden. Bei LAN muss ein TCP-Socket (oder ein UDP-Socket, nur für UDP-Streaming empfohlen) konfiguriert und darüber die Daten gesendet werden. Es können üblicherweise die selben Sockets benutzt werden, die zum Empfang der Daten vom RSR200 notwendig sind. Kommando-Pakete müssen immer mit genau der angegebenen Länge gesendet werden (bei USB 32 Bit Wort-weise, bei LAN Byte-weise).

Der RSR200 sendet Einzelpakete nur über TCP (Ausnahme: Versionsnummern bei Anforderung auch über UDP) und nur bei ausgeschaltetem Streaming. In allen anderen Fällen sind die Kommando-Daten in die Blöcke der IQ-Daten eingebettet.

#### 3.1 Handling der im Datenstrom eingebetteten Kommandos

Eingebettete Kommandos müssen aus den Datenblöcken extrahiert und bei Bedarf verarbeitet werden. Dabei existiert die Schwierigkeit, das beim Streaming ständig Daten übertragen und damit auch ständig Kommando-Daten vom RSR200 gesendet werden. Das PC-Programm muss jedoch erkennen können, ob diese Daten ein aktuell zu bearbeitendes Kommando, oder bereits verarbeitete alte Daten enthalten. Dazu enthält jeder Block eine Kommando-Nummer. Der Ablauf einer Kommando-Übertragung vom RSR200 an den PC erfolgt beim Streaming in folgender Weise:

- Ein Kommando wird vom RSR200 in den entsprechenden Kommando-Datenbereich der Blöcke geschrieben. Dieser Datenbereich wird bei jeder Sendung des Blocks an den PC gesendet. Damit kommen am PC immer die selben Kommando-Daten an, bis der RSR200 neue in den Datenbereich schreibt.
- Nach dem Schreiben von neuen Kommando-Daten setzt der RSR200 die 8 Bit Kommando-Nummer auf einen neuen Wert (üblicherweise fortlaufend um 1 erhöht, aber nicht garantiert). Die Kommando.Nr. startet nach Reset immer mit 0 (Kennzeichnung Neustart). Im weiteren Betrieb wird nie wieder die 0 verwendet (Überlauf 255 → 1). Ein neues (und damit vom PC zu verarbeitendes) Kommando wird als gültig bezeichnet, wenn das PC-Programm die neue Kommando-Nummer im aktuell empfangenen Block erkennt.
- In diesem Augenblick muss das Programm die zur neuen Nummer gehörenden Kommando-Daten aus dem Block extrahieren und (irgendwann) verarbeiten. Die Kommando-Daten bleiben so lange gültig, wie die Kommando-Nummer unverändert ist. Dies kann über viele Blöcke so bleiben, aber auch schon im nächsten Block kann ein neues Kommando mit neuer Nummer enthalten sein.

In einem USB-Block ist immer nur ein Kommando enthalten. Ein LAN-Block dagegen kann mehrere Kommandos enthalten. Die Zahl der im LAN-Block enthaltenen Kommandos wird im 32 Bit Feld „Kommando-Anzahl“ angegeben. Das PC-Programm muss die einzelnen Kommandos im Kommando-Block nacheinander bearbeiten, bis alle verarbeitet sind. Die Einhaltung der Reihenfolge kann wichtig sein!

Kommandos, die der PC an den RSR200 sendet, müssen ebenfalls immer eine Kommando-Nummer enthalten (siehe Abschnitt „Kommandos PC → RSR200“). Die Wahl dieser Nummer bleibt dem PC-Programm überlassen. Sie wird im RSR200 ausschließlich für die Erstellung der Bestätigungs-Kommandos verwendet. Sie ist dort zur Kennzeichnung vorhanden, für welchen empfangenen Befehl die Bestätigung erfolgt. **In selbst-generierten Kommandos sendet der RSR200 immer Nr. 0 an den PC!** Dies kann zur Erkennung selbst-generierter Kommandos verwendet werden, wenn der PC niemals 0 als Kommando-Nr. an den RSR200 schickt.

### 3.2 Kommandos RSR200 → PC

Vom RSR200 gesendeten Kommandos sind größtenteils Antworten auf Kommandos, die der PC vorher an den RSR200 gesendet hat (Bestätigungen über die Ausführung des Befehls oder angeforderte Daten). Der RSR200 kann auch selbst Kommandos generieren. Ein Kommando besteht aus einem Header von 4 Bytes, gefolgt von einem Befehls-Byte und einer genau bestimmten Anzahl von Datenbytes für diesen Befehl. Der Header ist folgendermaßen aufgebaut.

#### Kommando-Header: 4 Bytes

Byte-Nr.	Wert	Bedeutung
0	Signed Byte	Temperatur in °C
1	LSB GPS	Aktueller Wert der Frequenzkorrektur in Hz als vorzeichenbehafteter 14 Bit-Wert, OV1, OV2
2	MSB GPS, OV	
3	Kommando-Nr.	Nummer des aktuell übertragenen Kommandos

Das 1. Byte enthält den aktuellen Temperatur-Messwert (Kerntemperatur des FPGA im RSR200) als vorzeichenbehafteter 8 Bit-Wert (Bereich -128 ... +127°C).

Das 2. und 3. Byte enthält einen vorzeichenbehafteten 14 Bit-Wert (LSB Bit 0 bis Bit 7, MSB Bit 8 bis Bit 13) für die aktuelle Frequenzkorrektur des RSR200. Dabei existieren 2 Möglichkeiten (siehe Kommando „ADC-Takt Einstellen“ von PC an RSR200):

- Bei eingeschalteter interner Frequenzregelung gibt dieser Wert den aktuell notwendigen Korrekturwert (Stellgröße) der Regelung wieder. Er beschreibt, wie weit entfernt sich der ADC-Taktoszillator von der Sollfrequenz befindet und deshalb um diesen Wert korrigiert wird, damit die Sollfrequenz erzeugt wird. Die Auflösung beträgt 0,5 Hz / LSB.
- Bei ausgeschalteter Frequenzregelung gibt dieser Wert die Abweichung der aktuell erzeugten Frequenz von der Sollfrequenz an (IST – SOLL). Die Auflösung beträgt 0,1 Hz / LSB.

*Hinweis:* Die interne Frequenzmessung und -regelung benutzt als Zeitbasis den eingebauten GPS-Empfänger (1pps Signal). Ist kein GPS-Empfang möglich, so wird der höchstmögliche negative Wert ausgegeben (0x2000). Dieser Wert wird auch ausgegeben, wenn aus anderen Gründen kein gültiger Wert verfügbar ist (z. B. gerade Frequenz umgeschaltet, aber noch kein neuer Messwert vorhanden).

Die höchstwertigen 2 Bits des 3. Bytes kennzeichnen den Übersteuerungszustand der ADC (Bit 6: Kanal 1, Bit 7: Kanal 2). Ein gesetztes Bit bedeutet Übersteuerung, ein rückgesetztes Bit bedeutet Betrieb im linearen Bereich.

*Hinweis:* Die Messwerte für Temperatur und Frequenzkorrektur, sowie die Bits für die Übersteuerungsanzeigen werden so im Datenstrom aktualisiert, wie sie intern zur Verfügung stehen. Die Aktualisierungsrate steht in keinem festen Zusammenhang mit der ADC-Taktfrequenz, der Blockrate oder ähnlichen, von der Signalverarbeitung bestimmten, Zeiten.

Das 4. Byte enthält die Nummer des aktuell vom RSR200 gesendeten Kommandos. Sobald sich dieser Wert von einem zum folgenden Block ändert, ist in den weiteren Daten des Kommandos ein neuer Befehl enthalten. Blöcke mit gleicher Kommando-Nr. können, aber müssen nicht gleiche Daten / Befehle enthalten! Die **Änderung der Kommando-Nr. ist das Triggersignal** zur Auswertung der Kommandodaten des ersten Blocks mit der neuen Nummer. Der RSR200 sendet **nach Reset** zunächst die **Nummer 0** (kein Kommando verfügbar). Ab dem ersten Kommando werden immer Nummern ungleich 0 verwendet (üblicherweise aufwärts zählend mit Überlauf 255 → 1).

Dem Kommando-Header folgt eine bestimmte Zahl von Bytes, die den im Kommando enthaltenen Befehl mit den zugehörigen Daten enthalten. In einem USB-Block ist immer nur ein Kommando enthalten, es folgt unmittelbar dem Header. Kommandos werden über USB grundsätzlich nur im Stream-Betrieb gesendet.

In einem LAN-Block können mehrere Kommandos enthalten sein. Deshalb folgt dem Header zunächst ein 4 Byte-Feld, welches die Zahl der enthaltenen Kommandos als vorzeichenlosen 32 Bit-Wert angibt (siehe Datenstruktur LAN-Blöcke).

In einzelnen LAN-Paketen (nur Kommando „Meldung Versionsnummern LAN“ wenn kein Stream-Betrieb) ist immer nur ein Kommando enthalten und somit kein Zähler vor den Kommando-Bytes.



Nachfolgend die Struktur der verschiedenen Kommandos.

### Bestätigung: 8 Bytes

Byte-Nr.	Wert	Bedeutung
0	0x00	Befehl: Acknowledgement
1	0x00	
2	0x00	
3	0x00	
4	LSB empfangener Befehl	32 Bit Kommando-Nr. des empfangenen Befehls, dessen korrekte Ausführung hiermit bestätigt wird.
:	:	
7	MSB empfangener Befehl	

Allgemeine Bestätigung für die erfolgreiche / korrekte Ausführung eines vom PC gesendeten Kommandos, für das keine Rückmeldung von Daten erwartet wird.

### Spezial-Bestätigung: 8 Bytes

Byte-Nr.	Wert	Bedeutung
0	Befehl	Ausgeführter Befehl
1	1. Datenbyte	Rückgemeldete Daten
2	2. Datenbyte	
3	3. Datenbyte	
4	LSB empfangener Befehl	32 Bit Kommando-Nr. des empfangenen Befehls, dessen Verarbeitung hiermit bestätigt wird.
:	:	
7	MSB empfangener Befehl	

Spezielle Bestätigung von Kommandos, die eine Datenrückmeldung erwarten. Das 1. Byte enthält den im Kommando vom PC gesendeten Befehl. In den folgenden 3 Bytes sind weitere Informationen enthalten. Dies kann eine einfache „OK“ (Wert 0) oder „Nicht OK“ (Wert ungleich 0) im 1. Byte sein, oder die Rückmeldung verarbeiteter Daten (gleich den gesendeten Daten: Erfolgreich verarbeitet, ungleich den gesendeten Daten: Daten auf diesen Wert geändert).

*Hinweis:* Die Spezial-Bestätigung wird vom RSR200 auch für **selbst generierte Kommandos** verwendet (z. B. Meldung eines geänderten ADC-Taktes). In diesem Fall gibt es keine vorher vom PC gesendete 32 Bit PC-Kommando-Nr. Das Feld wird deshalb **mit Null gefüllt**.

### Meldung Versionsnummern USB: 8 Bytes

Byte-Nr.	Wert	Bedeutung
0	0x12	Befehl: Versionsnummern lesen
1	LSB S/N	24 Bit Seriennummer des Gerätes
2	:	
3	MSB S/N	
4	LSB Firmware	4 Digit hexadezimaler Wert Firmware-Version
:	:	
7	MSB Firmware	

Dieses Kommando wird vom RSR200 als Antwort auf die Anforderung zur Übermittlung der Versionsnummern über USB gesendet.

**Achtung!** USB-Pakete werden nur gesendet, wenn USB-Streaming eingeschaltet ist!

## Meldung Versionsnummern LAN: 12 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Länge	Länge des Befehls: 12
:	:	0
3	MSB Länge	0
4	0x12	Befehl: Versionsnummern lesen
5	LSB S/N	24 Bit Seriennummer des Gerätes
6	:	
7	MSB S/N	
8	LSB Firmware	4 Digit hexadezimaler Wert Firmware-Version
:	:	
11	MSB Firmware	

Dieses Kommando wird vom RSR200 als Antwort auf die Anforderung zur Übermittlung der Versionsnummern über LAN gesendet. Das Kommando wird immer als Einzelpaket über das Protokoll gesendet, über das die Aufforderung zur Sendung empfangen wurde (TCP oder UDP). Es enthält nicht die empfangene Kommando-Nummer und ist im Gegensatz zu den anderen Kommandos 12 Byte lang.

### 3.3 Kommandos PC → RSR200

Kommandos, die vom PC an den RSR200 gesendet werden, dienen der Einstellung des RSR200 auf verschiedene Betriebsarten. Außerdem ist die Übermittlung einer FPGA-Konfigurationsdatei („Firmware“) möglich, die alle Informationen zur Hardware-Konfiguration des FPGA und die Software der integrierten Prozessoren enthält.

Die Kommandos werden immer als einzelne Pakete über USB, TCP oder UDP versendet. Bei LAN werden nicht benutzte Bytes nicht gesendet (Paketlänge entsprechend verkürzt). Bestätigungen werden vom RSR200 immer über die Schnittstelle zurück gesendet, über die das Kommando empfangen wurde. Folgende Kommandos kann der RSR200 verarbeiten.

#### Kommando Reset: 8 bytes

Byte no.	Value	Description
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0xB2	Befehl: Reset
5	0x00	
6	0x00	
7	0x00	

Dieses Kommando führt einen Neustart aus. Der RSR200 lädt die Firmware aus dem Speicher und startet die Ausführung der Software so wie nach dem Einschalten des Geräts.

#### Kommando Versionsnummern Lesen: USB 8 Bytes, LAN 6 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0x12	Befehl: Versionsnummern lesen
5	Byte	Zähler Wiederholungen
6	n/a	keine Bedeutung / bei LAN nicht übertragen
7	n/a	keine Bedeutung / bei LAN nicht übertragen

Der RSR200 antwortet darauf mit der Ausgabe der Versionsnummern des Gerätes (Kommando „Meldung Versionsnummern“).

### Kommando Stream Starten: USB 8 Bytes, LAN 7 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0x15	Befehl: Streaming der Daten starten
5	Port	Schnittstelle: 0 = UDP, 1 = TCP, 2 = USB
6	Size	Blocklänge bei LAN: 7 = 1-Kanal 16 Bit, 15 = 2-Kanal 16 Bit, andere Werte = 1-Kanal 24 Bit
7	n/a	keine Bedeutung / bei LAN nicht übertragen

Das Kommando startet den Stream-Betrieb (fortlaufendes Senden von Blöcken) für die angegebene Schnittstelle. Die geforderte Blocklänge bei LAN muss der aktuellen Einstellung des RSR200 entsprechen (Auflösung und Kanalzahl, siehe Kommando „Schnittstelle Einstellen“).

Dieses Kommando wird vom RSR200 nicht mit einem Bestätigungs-Kommando quittiert.

*Hinweis:* USB-Streaming wird vom RSR200 automatisch nach jedem Einschalten / Reset gestartet.

*Hinweis:* Werden Auflösung und / oder Kanalzahl der LAN-Schnittstelle nach dem Start des Streams geändert (Kommando „Schnittstelle Einstellen“), ist ein neues Kommando zum Stream-Start notwendig.

### Kommando Stream Anhalten: USB 8 Bytes, LAN 7 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0x16	Befehl: Streaming der Daten stoppen
5	Port	Schnittstelle: 0 = UDP, 1 = TCP, 2 = USB
6	Byte	Zähler Wiederholungen
7	n/a	keine Bedeutung / bei LAN nicht übertragen

Der Stream-Betrieb der angegebenen Schnittstelle wird angehalten.

Dieses Kommando wird vom RSR200 nicht mit einem Bestätigungs-Kommando quittiert.

**Achtung!** Das Beenden des Streamings über die USB-Schnittstelle schließt den Transmitter-Endpoint komplett! Es können keine Daten oder Kommandos mehr vom RSR200 an den PC gesendet werden! Der Empfang über USB (z. B. erneuter Stream-Start) ist weiterhin möglich.

### Kommando ADC-Takt Einstellen: 8 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0xF2	Befehl: Wert für ADC-Takt setzen
5	ADC-CLK	LSB ADC-Takt in 0,1 MHz
6	Bit 0..6: ADC-CLK Bit 7: GPS-Dis	MSB ADC-Takt GPS-Frequenzregelung ausschalten
7	Byte	Zähler Wiederholungen

Das Kommando stellt die Taktfrequenz der ADC-Schaltkreise ein. Die gewünschte Taktfrequenz muss im möglichen Bereich des Gerätes liegen (RSR200B: 70,0 ... 200,0 MHz = ADC-Takt 700 ... 2000).

*Hinweis:* Jede Änderung der ADC-Takteinstellung erzeugt ein Synchronisations-Ereignis (Reset der DDS-Generatoren in den Mischoszillatoren und der FIR-Filter und Ablaufsteuerungen in den Dezimierern). Dadurch wird immer die gleich Phasenlage der Signale in Kanal 1 und 2 gewährleistet.

**Achtung:** Wenn das MSB für ADC-Takt Null ist, wird das LSB als Takt-Wert mit 1 MHz Auflösung bewertet (Bereich 70 ... 200).

Das Bit „GPS-Dis“ gibt die Art der Frequenzregelung an: Wert = 0: Interne Regelung ein, anderer Wert: Regelung aus.

Dieses Kommando wird mit einer speziellen Bestätigung quittiert. Im 1. Datenbyte der Bestätigung wird das LSB der tatsächlich eingestellten Taktfrequenz angegeben, im 2. Byte das MSB und das Bit für den Wert GPS-Dis.

*Hinweis:* Bei eingeschalteter Regelung wird vom RSR200 der interne Korrekturwert mit 0,5 Hz Auflösung ausgegeben, bei ausgeschalteter Regelung die aktuelle Abweichung IST - SOLL in 0,1 Hz Auflösung.

**Achtung:** Ab Firmware V223 überwacht der RSR200 die Temperatur. Bei Überschreitung von 87°C wird schrittweise (ca. alle 8 s) der ADC-Takt um 1 MHz vermindert, bis der Temperatur-Messwert wieder 87°C oder kleiner beträgt. Der RSR200 meldet die verminderte Taktfrequenz mit der zugehörigen speziellen Bestätigung. Die Bestätigung enthält die zurückgemeldete Kommando-Nr. 0 (Hinweis, dass diese Bestätigung vom RSR200 selbst generiert wurde und keine Antwort auf ein Kommando ist).

#### Kommando Frequenzgeneratoren oder IP-Adresse Einstellen: USB 12 Bytes, LAN 11 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0xB0	Befehl: Wert für Generatoren / IP setzen Einstellwert: 0 = LO-Frequenz Kanal 1 1 = LO-Frequenz Kanal 2 2 = LO-Frequenz beide Kanäle 9 = Magnitude und Phase Kanal 2 10 = IP-Adresse 32 Bit Binärformat andere Werte: keine Wirkung
5	Channel	
6	LSB Daten	
7	(MSB)	
8	(LSB)	
9	MSB Daten	MSB 32 Bit-Wert oder MSB 16 Bit Phase
10	Byte	Zähler Wiederholungen
11	n/a	keine Bedeutung / bei LAN nicht übertragen

Das Kommando stellt die Mischoszillatoren einer oder beider Kanäle ein. Es dient außerdem zur Übermittlung einer neuen IP-Adresse.

Dieses Kommando wird nach Ausführung mit einer speziellen Bestätigung quittiert. Das 1. Datenbyte enthält den Parameter „Channel“ (Byte 5), das 2. Byte die Erfolgsmeldung (0 = OK, anderer Wert = nicht ausgeführt).

**Achtung!** LO-Frequenz, Phase und Magnitude für Kanal 2 werden nur ausgewertet, wenn Kanal 2 eingeschaltet ist (vorher auf Betriebsart „Diversity“ oder „Separate“ geschaltet wurde).

*Hinweise:* Frequenzen werden als vorzeichenbehafteter 32 Bit Wert mit 1 Hz Auflösung interpretiert. Es liegt in der Verantwortung des Senders, geeignete Werte zu übermitteln. Der RSR200 berechnet intern die notwendigen Daten zur Einstellung der Generatoren. Überläufe (z. B. Frequenz höher als maximale ADC-Taktfrequenz) werden ohne Korrektur in die Generatoren programmiert.

Die Magnitude zur Beeinflussung des 2. Kanals im Diversity-Betrieb wird als vorzeichenloser 16 Bit Wert interpretiert (max. 0xFFFF = 8-1 LSB fache Verstärkung).

Die Phase zur Beeinflussung des 2. Kanals im Diversity-Betrieb wird als vorzeichenbehafteter 16 Bit Wert interpretiert (0x8000 = -180° ... 0x7FFF = +180°-1 LSB).

Der 32 Bit Wert für eine IPv4-Adresse (z. B. mit der Funktion inet\_addr() aus der Textform gewonnen) wird nichtflüchtig im RSR200 gespeichert. Er wird ab dem nächsten Einschalten / Reset als Standard-Adresse verwendet.

**Kommando Variablen 16 Bit-Wert Einstellen: USB 12 Bytes, LAN 9 Bytes**

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0xF5	Befehl: Wert für 16 Bit Variable setzen
5	Variable Nr.	Nummer der zu ändernden Variablen
6	LSB Daten	LSB 16 Bit-Wert für Variable
7	MSB Daten	MSB 16 Bit-Wert für Variable
8	Byte	Zähler Wiederholungen
9	n/a	keine Bedeutung / bei LAN nicht übertragen
10	n/a	keine Bedeutung / bei LAN nicht übertragen
11	n/a	keine Bedeutung / bei LAN nicht übertragen

Das Kommando dient zur Einstellung verschiedener Werte, die mit einem 16 Bit (2 Byte) großen Datenwort dargestellt werden können. Nachfolgend die möglichen Einstellungen.

Variable Nr.	Bedeutung	Zulässiger Wertebereich
0	CLK-Frequenzkorrektur in Hz	-3276,8 .. +3276,7 Hz
1	Abschwächer ADC1	LSBits: 0 ... 35 (-7 .. +28 dB), Bit 7 = 1: ADC2 ebenso
2	Abschwächer ADC2	LSBits: 0 ... 35 (-7 .. +28 dB), MSBits: 0
3	Antennensteuerung HF1/VHF	LSB: 8 Bit UART-Datenwort für Antenne (siehe Steuergeräte)
4	Antennensteuerung HF2	MSB: 0
5	Schalter	LSB: Bits für Schalter, MSB: 0 Bit 0: 0 = ADC2 CLK in Phase, 1 = ADC2 CLK invertiert Bit 1: 0 = ADC1 an HF1, 1 = ADC1 an VHF Bit 2: 0 = ADC2 an ADC1 (parallel), 1 = ADC2 an HF2 Bit 3: Fernspeisung HF1/VHF, 0 = aus, 1 = ein Bit 4: Fernspeisung HF1/VHF, 0 = +12 V, 1 = Steuerung Bit 5: Fernspeisung HF2, 0 = aus, 1 = ein Bit 6: Fernspeisung HF2, 0 = +12 V, 1 = Steuerung Bit 7: Vorverstärker VHF, 0 = aus, 1 = ein
6	Antennensteuerung HF1/VHF	LSB 16 Bit UART-Datenwort für Frequenz (siehe Steuergeräte) +
7	Antennensteuerung HF2	MSB 16 Bit UART-Datenwort für Frequenz (siehe Steuergeräte)
8 .. 255	n/a	nicht definiert

Dieses Kommando wird vom RSR200 mit einer speziellen Bestätigung quittiert. Das 1. Datenbyte der Bestätigung enthält die Variablennummer, das 2. (LSB) und 3. (MSB) den tatsächlich im RSR200 verwendeten 16 Bit Wert (evtl. auf zulässige Werte geändert).

**Kommando Datenübertragung Einstellen: USB 12 Bytes, LAN 9 Bytes**

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0xB4	Befehl: Schnittstelle einstellen
5	Schnittstelle	1: USB, 2: LAN, 3: nur DSP einstellen, andere: nicht definiert
6	Port Mode	Byte für Port-Modus, bei Schnittstelle = 3: 0
7	DSP Mode	Byte für DSP-Modus
8	Byte	Zähler Wiederholungen
9	n/a	keine Bedeutung / bei LAN nicht übertragen
10	n/a	keine Bedeutung / bei LAN nicht übertragen
11	n/a	keine Bedeutung / bei LAN nicht übertragen

Das Kommando erlaubt die Einstellung einer Schnittstelle hinsichtlich der verwendeten Datenraten, Auflösungen und Kanalzahlen. Folgende Möglichkeiten existieren für die Werte Port Mode und DSP Mode.

## 1 Byte Port-Modus

Bitnummer	Bedeutung	Zulässiger Wertebereich
0	3 Bit Dezimierungsrate D	0 (Rate = 2) .. 5 (Rate = 64), andere unzulässig, Dezimierungsrate = $2^{\exp(D+1)}$
1		
2		
3	Kanalwahl	Wenn Bit 4 = 0: 0 = Daten von ADC 1, 1 = von ADC 2 Wenn Bit 4 = 1: 0 = ADC1 an Kanal 1. ADC2 an Kanal 2 1 = ADC1 an Kanal 2, ADC2 an Kanal 1
4	Kanalzahl	0 = 1-Kanal, 1 = 2-Kanal
5	Bitbreite IQ-Daten	0 = 24 Bit, 1 = 16 Bit (muss immer bei 2-Kanal)
6	n/a	nicht definiert
7	n/a	nicht definiert

## 1 Byte DSP-Modus

Bitnummer	Bedeutung	Zulässiger Wertebereich
0	2 Bit Betriebsart	0 = ADC1 und ADC2 unabhängig (Portmode Bit 4 muss 1 sein!)
1		1 = ADC1 + ADC2 parallel (Addition Daten)
		2 = ADC1 + ADC2 seriell (CLK ADC2 muss invertiert sein!)
		3 = ADC1 + ADC2 parallel (Diversity)
2	n/a	0
3	Seitenband	0 = unteres Seitenband bei seriell, 1 = oberes Seitenband
4	n/a	0
5	n/a	0
6	n/a	0
7	n/a	0

Dieses Kommando wird vom RSR200 mit einer speziellen Bestätigung quittiert. Das 1. Datenbyte der Bestätigung enthält die Quittung. Bei Quittung 0 läuft die Schnittstelle sofort mit den geänderten Einstellungen weiter. Bei Quittung ungleich 0 muss die Schnittstelle geschlossen und neu initialisiert und verbunden werden.

*Hinweis:* Jede Umschaltung von Einstellungen der Schnittstellen erzeugt ein Synchronisations-Ereignis (Reset der DDS-Generatoren in den Mischoszillatoren und der FIR-Filter und Ablaufsteuerungen in den Dezimierern). Dadurch wird immer die gleich Phasenlage der Signale in Kanal 1 und 2 gewährleistet.

**Achtung!** Beim Umschalten des Modus für die LAN-Schnittstelle stoppt der RSR200 selbständig den Stream-Betrieb (falls eingeschaltet) und setzt die Schnittstelle auf die angeforderten Daten. Der Stream-Betrieb muss danach mit einem separaten Kommando „Start Stream“ mit Angabe einer zu diesen Einstellungen passenden Blockgröße wieder neu gestartet werden. Solange der Stream ausgeschaltet ist, kann keine Bestätigung empfangen werden (sie ist ab dem ersten Block des neuen Streams enthalten).

## 3.4 Firmware Update

**Achtung!** Änderungsmöglichkeiten an der Firmware des Gerätes sollten nur vorgesehen werden, wenn jederzeit die Möglichkeit besteht, die Original-Firmware herstellermäßig neu einprogrammieren zu können. Dazu ist ein „Xilinx Platform Cable USB II“ Emulator mit Anschluss an den internen Programmier-Anschluss des RSR200, sowie die Xilinx-Entwicklungsumgebung und eine Original Firmware-Datei („Golden Image“) des RSR200-Herstellers erforderlich!

Die Firmware des RSR200 besteht aus der Hardware-Konfiguration des FPGA und der Software für die im FPGA integrierten Prozessoren. Beides kann gemeinsam über eine einzige Binärdatei in den RSR200-internen Flash-Speicher neu einprogrammiert werden.

Im Flash-Speicher existieren immer 2 Abbilder dieser Binärdatei:

- Löschbares und über eine Schnittstelle neu ladbares Versions-Image (Firmware, die nach dem Einschalten / Reset geladen und ausgeführt wird).
- Nur über den internen Programmier-Anschluss änderbares „Golden Image“ (Reserve-Datei, falls Versionsdatei nicht vorhanden oder beschädigt ist).

Das „Golden Image“ wird beim Hersteller fest eingebrannt und. Im Grunde ist es eine ältere, gut geprüfte und sicher laufende Version der Firmware. Sie ermöglicht den Start des Gerätes, wenn keine gültige Versions-Datei geladen ist (Version des „Golden Image“ prüfen und zugehörige PC-Software verwenden!).

Das Versions-Image (üblicherweise Erweiterung „.Bit“) kann über die Datenverbindung vom PC gelöscht und neu eingespeichert werden. Dazu stehen folgende Kommandos zur Verfügung.

#### Kommando Freigabe Firmware-Update: USB 12 Bytes, LAN 9 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0x0B	Befehl: Freigabe Update
5	LSB S/N	Seriennummer des RSR200
6	:	
7	MSB S/N	
8	Byte	Zähler Wiederholungen, wird nicht ausgewertet
9	n/a	keine Bedeutung / bei LAN nicht übertragen
10	n/a	keine Bedeutung / bei LAN nicht übertragen
11	n/a	keine Bedeutung / bei LAN nicht übertragen

Die mit Kommando „Meldung Versionsnummern“ vom RSR200 ausgegebene Seriennummer des Gerätes muss angegeben werden.

Der RSR200 quittiert dieses Kommando mit der speziellen Bestätigung und Zurückgabe des Befehls und der Seriennummer (in den 3 Datenbytes). Daraufhin ist der Update-Vorgang freigeschaltet und kann mit dem Kommando „Start Firmware-Update“ beginnen. Jede andere Rückmeldung (andere Seriennummer) zeigt an, dass kein Update möglich ist.

*Hinweis:* Jedes andere diesem Kommando folgende Kommando außer „Start Firmware-Update“ sperrt den Update-Vorgang wieder.

#### Kommando Start Firmware-Update: USB 12 Bytes, LAN 9 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0x0C	Befehl: Start Update
5	LSB Länge	Länge der Bit-Datei in Bytes
6	:	
7	MSB Länge	
8	Byte	Zähler Wiederholungen, wird nicht ausgewertet
9	n/a	keine Bedeutung / bei LAN nicht übertragen
10	n/a	keine Bedeutung / bei LAN nicht übertragen
11	n/a	keine Bedeutung / bei LAN nicht übertragen

Es ist die Länge der zur übertragenden Binärdatei in Bytes anzugeben.

Nach Empfang dieses Kommandos löscht der RSR200 die Versionsdatei. Wenn der Löschvorgang beendet ist, sendet er die spezielle Bestätigung mit 0 = OK oder 4 = Fehler im 1. Datenbyte der Bestätigung. Bei OK erwartet er ab diesem Zeitpunkt die Daten der Versions-Datei. Bei Fehlern wird der Vorgang abgebrochen.

**Achtung!** Nach einem abgebrochenen Firmware Update ist der Zustand der Versions-Datei undefiniert. Ein Neustart des RSR200 könnte mit Start des „Golden Image“ gelingen, oder in der beschädigten

Versions-Datei hängenbleiben. Abbrüche des Updates sollten deshalb immer mit erfolgreicher Löschung oder Übertragung des Versions-Image enden. Gegebenenfalls das Update oder zumindest die Löschung vor Reset des Gerätes neu starten und erfolgreich ausführen!

#### Kommando Daten für Firmware-Update: USB 264 Bytes, LAN 263 Bytes

Byte-Nr.	Wert	Bedeutung
0	LSB Kommando-Nr.	32 Bit Nummer dieses Kommandos
:	:	
3	MSB Kommando-Nr.	
4	0x0D	Befehl: Daten Update
5	Paketnummer	Fortlaufender 7 Bit-Zähler Pakete, Bit 7 immer 1
6	1. Byte des Pakets	Fortlaufende Daten aus Bit-Datei
:	:	
261	256. Byte des Pakets	
262	Byte	Zähler Wiederholungen
263	n/a	keine Bedeutung / bei LAN nicht übertragen

Dieses Kommando überträgt einen Block von 256 Bytes der Binärdatei. Der RSR200 speichert die Daten im Flash und quittiert mit der speziellen Bestätigung.

Im 1. Datenbyte der Bestätigung ist die Paketnummer (= Paket OK) oder der Fehlercode 4 (Fehler bei Übertragung / Speicherung, Abbruch!) oder Code 0 (letztes Paket, alles OK) enthalten. Im 2. (LSB) und 3. (MSB) Datenbyte ist eine 16 Bit Checksumme enthalten. Diese wird durch fortlaufende binäre Addition aller bisher eingegangener Bytes der Binärdatei generiert (Überlauf 16 Bit abgeschnitten). Sie kann im PC gegen eine ebenso generierte Checksumme geprüft werden, um die Datenintegrität zu gewährleisten.

#### Achtung! Siehe Kommando „Start Firmware Update“.

Die Übertragung der Bytes der Firmware muss am Ende der Datei starten und rückwärts bis zum Anfang erfolgen! Der RSR200 erwartet immer volle Pakete mit 256 Bytes (innerhalb des Pakets aufsteigende Reihenfolge der Bytes). Deshalb muss das erste Paket am Ende mit Leerbytes (0xFF) aufgefüllt werden, wenn die Binärdatei nicht genau ein Vielfaches von 256 Bytes lang ist. Das letzte Paket muss das 1. bis 256. Byte der Binärdatei enthalten.

Die Bestätigungen für die Pakete werden ganz normal im Daten-Stream als Kommandos gesendet. Deshalb muss der Stream-Betrieb eingeschaltet sein. Um den PC von unnötigen Arbeiten zu entlasten (und damit die Sicherheit der Datenübertragung zu erhöhen), sollte keine Verarbeitung der IQ-Daten während des Updates erfolgen.

Nach erfolgreichem Update arbeitet der RSR200 normal weiter. Erst nach Neustart (Einschalten / Reset) wird die neue Versions-Datei geladen und ausgeführt.

### 3.5 Kommando-Wiederholungen

In den Kommandos PC → RSR200 ist ein Byte mit der Bezeichnung „Zähler Wiederholungen“ vorhanden. Es ist standardmäßig mit dem Wert 0 besetzt.

Zur Erhöhung der Übertragungssicherheit kann das PC-Programm prüfen, ob ein an den RSR200 gesendetes Kommando innerhalb einer akzeptablen Zeit ausgeführt wurde oder nicht (Verlust des Kommando-Pakets, Fehler im RSR200, Verlust Bestätigung, ...). Trifft eine erwartete Bestätigung für ein Kommando nicht innerhalb angemessener Frist ein, so kann das Kommando ein- oder mehrmals wiederholt werden. Für ein wiederholtes Kommando sollte die Kommando-Nr. beibehalten und nur der Wiederholungs-Zähler erhöht werden.

Der RSR200 kann beim Empfang wiederholter Kommandos prüfen, ob er dieses Kommando schon einmal ausgeführt hat (und nur der PC die Bestätigung nicht erhalten hat). In diesem Fall wird das Kommando nicht noch einmal ausgeführt, es wird nur eine neue Bestätigung versendet.

Hinweis: In der Firmware Version 22x prüft der RSR200 den Wiederholungszähler nicht. Es sollten keine Kommando-Wiederholungen, sondern neue Kommandos gesendet werden. Die Folgen etwaiger wiederholter Ausführung gleicher Kommandos müssen einkalkuliert werden.



## 4. Hinweise zum Betrieb des RSR200

Nach dem Einschalten oder einem Reset (Software-Reset oder schwerer Fehler) befindet sich der RSR200 in folgendem Grundzustand:

- ADC CLK ist unbestimmt, liegt bei ca. 125 MHz
- Betriebsart: Parallel (ADC1 + ADC2)
- Dezimierungsraten: 16
- Ports: 1-Kanal 24 Bit, Daten an USB (in Stream-Mode geschaltet)
- Einstellbare Werte (Frequenzen, Abschwächer usw.) auf Null bzw. Minimalwert
- Eingänge: HF1 an ADC1 und ADC2, Fernspeisungen / Antennensteuergeräte aus
- Frequenzkorrektur per GPS aktiv

Außer der IP-Adresse wird kein Einstellwert im RSR200 fest gespeichert.

### 4.1 Verwendung der USB-Schnittstelle

Das Auffüllen der Pufferspeicher der USB-Schnittstelle beginnt unmittelbar nach der Initialisierung des Gerätes. Das Kommandofeld ist mit Kommando-Nr. 0 initialisiert und die Kommandofelder sind unbestimmt. Ab Verfügbarkeit von Messwerten (Temperatur und GPS) werden diese fortlaufend in den Block geschrieben. Eine Änderung dieser Werte ist kein Kommando und führt damit nicht zur Änderung der Kommando-Nr.

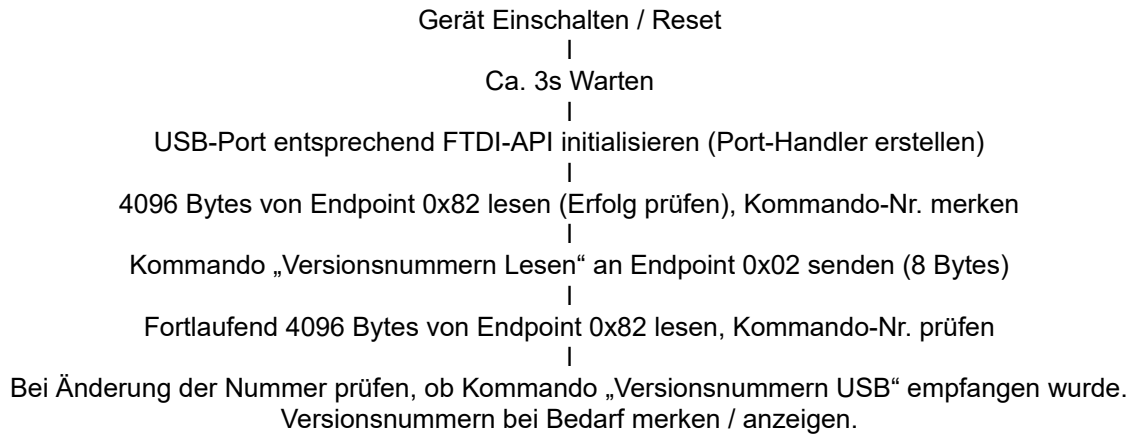
Jede Änderung der Kommando-Nummer repräsentiert die Übertragung eines neuen Kommandos. Die Daten des aktuellen Kommandos sind so lange gültig (über beliebig viele Pakete), bis eine neue Kommando-Nummer gesendet wird.

Um eine Verbindung vom PC zum RSR200 herzustellen, bietet die FTDI-API verschiedene Möglichkeiten. Das folgende Bild zeigt die Konfiguration des FT601Q. Für eine Verwendung der Seriennummer kann diese wahlfrei geändert werden (vorzugsweise auf die Seriennummer des RSR200). **Alle anderen Parameter dürfen nicht geändert werden!**

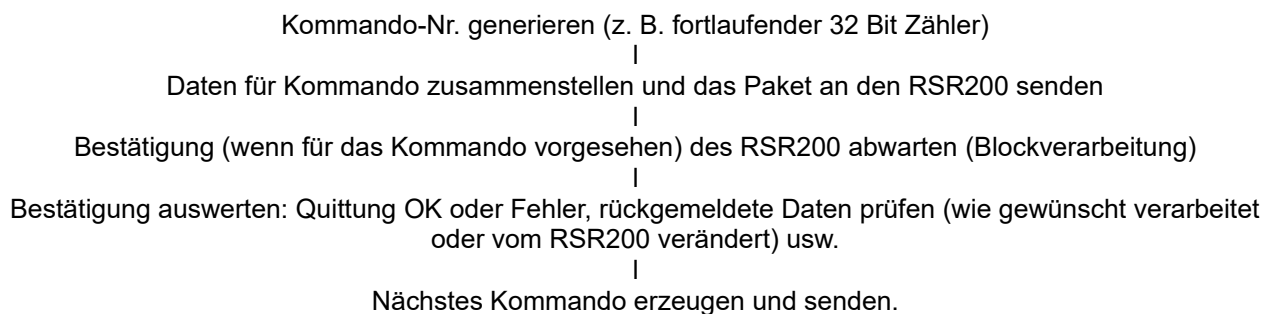
The screenshot shows the 'FT60X Chip Configuration Programmer' window. It is divided into several sections:

- Device Descriptor:** Vendor ID (0x0403), Product ID (0x601F).
- String Descriptor:** Manufacturer (FTDI), Description (FTDI SuperSpeed-FIFO Bridge), Serial Number (000000000028) with a 'Randomize' button.
- Configuration Descriptor:** Bus-powered (unselected), Self-powered (selected), Remote wakeup (unselected), Max Power (mA) (96).
- Pin Drive Strength Control:** FIFO Data (50 Ohm), GPIO0 (50 Ohm), FIFO Clock (50 Ohm), GPIO1 (50 Ohm).
- Interrupt Latency:** bInterval (9).
- Data Transfer:** FIFO Clock (100 MHz), FIFO Mode (245 Mode), Channel Config (1 Channel).
- Optional Feature Support:** Battery Charging Enabled (unselected), DCP (11 (GPIO1=1, GPIO0=1)), CDP (10 (GPIO1=1, GPIO0=0)), SDP (01 (GPIO1=0, GPIO0=1)), Default (00 (GPIO1=0, GPIO0=0)), Ignore Session Underrun (unselected) with sub-options 'On Multiple of MaxPacketSize bytes' and 'On Multiple of FIFO bus-width bytes'.
- Notification Message Enabled:** CH1, CH2, CH3, CH4 (all unselected).

Zur Benutzung der USB-Schnittstelle wird folgender Ablauf empfohlen.



Aber hier können die IQ-Daten der USB-Blöcke sicher verwendet werden. Dazu muss das Programm ständig alle Daten vom RSR200 empfangen und entsprechend der Blockstruktur verarbeiten. Kommandos mit den gewünschten Einstellungen können davon unabhängig an den RSR200 gesendet werden. Dabei empfiehlt sich folgender „Ping-Pong“ Ablauf von Kommandos und Bestätigungen:



Kommandos, für die keine einzelne Bestätigung ausgewertet werden soll (wird aber immer empfohlen!), können mit einer gemeinsamen Kommando-Nr. als Kommando-Block gesendet werden. Die einzelnen Kommandos müssen in den Sendedaten lückenlos hintereinander liegen. Aufgrund der fest vorgegebenen Kommandolängen kann der RSR200 alle Kommandos nacheinander lesen und verarbeiten. Die maximale Blocklänge beträgt 264 Bytes.

Nach Senden eines Kommandoblocks muss auf die Bestätigung des letzten Kommandos (welches eines mit Anforderung einer Bestätigung sein muss) gewartet werden, bevor neue Kommandos gesendet werden können.

## 4.2 Verwendung der LAN-Schnittstelle mit TCP

Nach dem Einschalten / Reset versucht der RSR200 ca. 5 s lang, von einem DHCP-Server eine IP-Adresse (IPv4) zu beziehen. Geschieht dies erfolgreich, wird für die Einschaltdauer des Gerätes diese Adresse verwendet. Wenn nicht, so wird die im Gerät gespeicherte Adresse verwendet (Werkseinstellung 192.168.1.10).

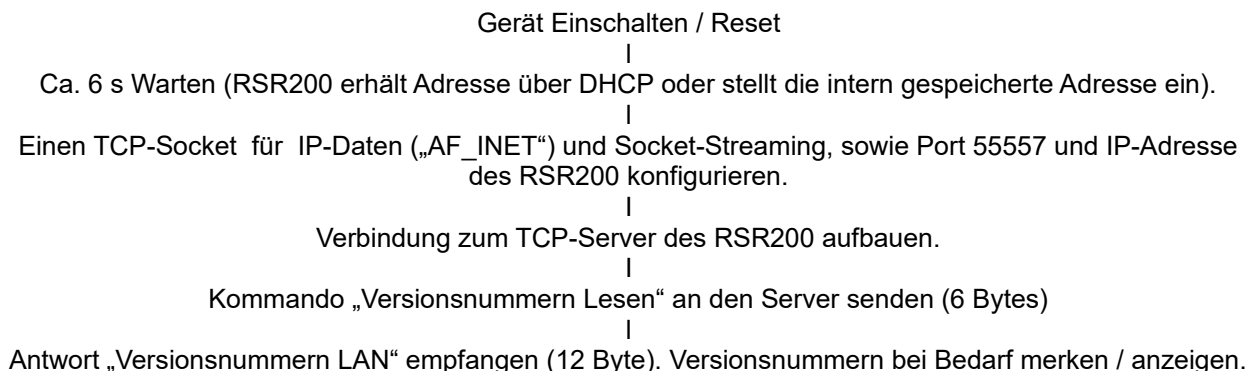
Der RSR200 konfiguriert einen TCP-Server, der auf Port 55557 auf eingehende Verbindungen wartet. Eine gültige Anfrage führt zum Aufbau der Verbindung. Es ist nur eine Verbindung zu einem Client möglich. Sobald die Verbindung etabliert ist, kann der Client Kommandos an den RSR200 senden. Solange der TCP-Server nicht auf Stream-Betrieb geschaltet ist, antwortet er mit einzelnen Paketen nur auf die Anfrage zur Ausgabe der Versionsnummern LAN.

Nach Empfang des Kommandos „Stream Starten“ für TCP beginnt der Server mit dem Senden von LAN-Blöcken. Das Kommando muss die korrekte Größe der Blöcke entsprechend der Einstellung der Schnittstelle (vorher per Kommando „Schnittstelle Einstellen“ ausführen) angeben. Der Client kann die Blöcke in der gesamten Größe empfangen und die enthaltenen Daten und Kommandos verarbeiten.

Die Blöcke müssen auf korrekte Datenausrichtung an den Blockgrenzen geprüft und bei Bedarf korrekt ausgerichtet werden. Dazu können die in jedem Block vorhandenen Blockzähler und Synchronisations-Daten verwendet werden.

Am Ende jedes Blocks ist ein von der Blockgröße abhängiger Platz für Kommandos vom RSR200 zum TCP-Client vorhanden. Darin können Kommandoblöcke verschiedener Größe vorhanden sein. Die Zahl der enthaltenen Kommandos ist im Block angegeben. Der Client muss den Block auswerten, sobald er eine Änderung der Kommando-Nummer erkennt. Die Daten (Kommandos) des Blocks sind nur für diesen Block gültig. Folgende Blöcke mit gleicher Kommando-Nummer können bereits andere (neue, noch ungültige / nicht vollständige) Daten enthalten!

Zur Benutzung der LAN-Schnittstelle mit TCP wird folgender Ablauf empfohlen.



Der PC kann nun über den TCP-Socket weitere Kommandos an den RSR200 senden. Um Daten vom RSR200 zu empfangen, muss das Kommando „Stream Starten“ für TCP gesendet werden. Vorher muss die Schnittstelle mit Kommando „Datenübertragung Einstellen“ konfiguriert werden. Danach kann die Verarbeitung der Blöcke beginnen. Die Kommandoverarbeitung sollte wie bei USB beschrieben nach dem „Ping-Pong-Prinzip“ erfolgen (Kommando → Bestätigung → nächstes Kommando...). Wenn Befehle aufgrund der Samplingrate schneller gesendet werden, als ein LAN-Block empfangen werden kann, werden die Bestätigungen im nächsten LAN-Block gebündelt („Kommando-Anzahl“ ist größer als 1).

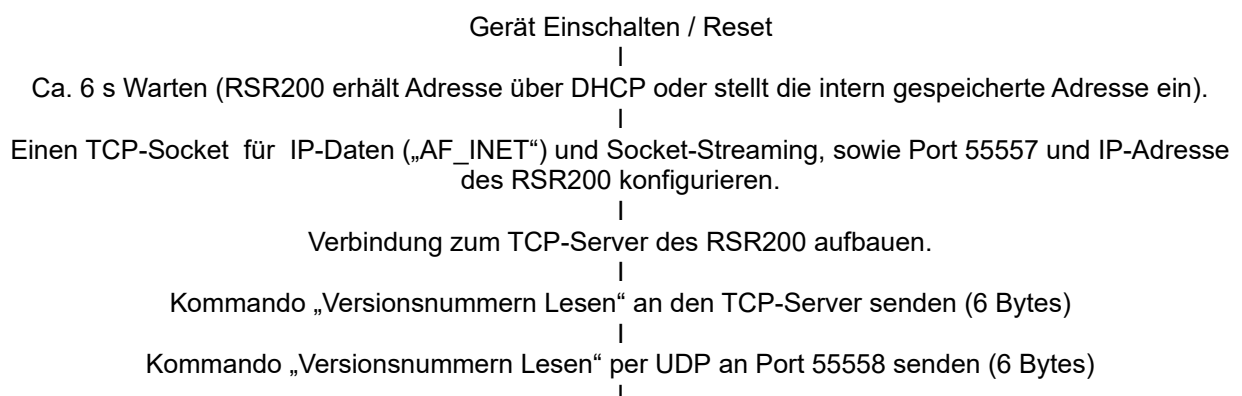
Der Stream-Betrieb kann jederzeit durch Kommando „Stream Anhalten“ (7 Bytes) unterbrochen werden. Beim erneuten Starten mit „Stream Starten“ muss die korrekte Blockgröße verwendet werden (zwischenzeitlich evtl. geändert), sonst werden vom RSR200 Blöcke mit falscher Länge gesendet.

### 4.3 Verwendung der LAN-Schnittstelle mit UDP

Die Datenübertragung mit UDP-Paketen ermöglicht unter Umständen eine höhere Datenrate (kein Empfang von ACKs vom PC notwendig, keine Stream-Unterbrechungen bei fehlerhaften Paketen). Dafür können Datenfehler bzw. -verluste auftreten, die nicht erkannt oder korrigiert werden können. UDP-Betrieb wird deshalb nur für die Datenrichtung RSR200 → PC empfohlen. Der RSR200 sendet Kommandos in jedem Block, wobei ein Kommando üblicherweise viele Male wiederholt wird. Ein kompletter Verlust eines Kommandos ist damit sehr unwahrscheinlich. Geringe Verluste von IQ-Daten sind oft akzeptabel.

Der RSR200 kann unter seiner aktuellen IP-Adresse (siehe Konfiguration TCP) auf Port 55558 UDP-Pakete empfangen. Dies wird nur zur Initialisierung der Verbindung mittels Kommando „Versionsnummern Lesen“ empfohlen. Der RSR200 entnimmt dem UDP-Paket die Daten des Senders und sendet fortan UDP-Pakete (sofern als Stream freigegeben) immer an diesen UDP-Partner. Alle weiteren Kommandos während des laufenden Betriebs sollten vom PC über TCP gesendet werden.

Zur Benutzung der LAN-Schnittstelle mit UDP wird folgender Ablauf empfohlen.



Der PC kann nun über UDP oder vorzugsweise TCP (Sicherheit!) weitere Kommandos an den RSR200 senden. Um Daten per UDP vom RSR200 zu empfangen, muss das Kommando „Stream Starten“ für UDP als UDP-Paket gesendet werden. Vorher muss die Schnittstelle mit Kommando „Datenübertragung Einstellen“ konfiguriert werden. Danach kann die Verarbeitung der UDP-Daten beginnen. Die UDP-Pakete sollten einzeln empfangen und zum endgültigen LAN-Block zusammengesetzt werden.

Der UDP-Streambetrieb kann jederzeit durch Kommando „Stream Anhalten“ (7 Bytes) abgebrochen werden. Ein erneuter Start des Streams kann sowohl mit TCP, als auch mit UDP erfolgen.

#### 4.4 Antennensteuerung

Der RSR200 enthält 2 Steuergeräte RSW zur Ansteuerung von Aktivantennen. Das Steuergerät für Kanal 1 wird entweder an Eingang HF1 oder an Eingang VHF geschaltet (wenn aktiviert), abhängig von der Schaltung des Kanals an den jeweiligen Eingang. Steuergerät 2 ist immer an HF2 geschaltet (wenn aktiviert).

Jedes Steuergerät kann entweder eine feste Spannung von 12 V ausgeben (reine Fernspeisung von beliebigen Antennen), oder eine RS-232 modulierte Spannung zur Steuerung spezieller Antennen. Die Daten zur Steuerung der Antennen können mittels Kommando „Variablen 16 Bit-Wert Einstellen“ übergeben werden. Je nach zu steuernder Antenne müssen die Daten dem von der Antenne erwarteten Format und Wertebereich entsprechen. Dazu stehen verschiedene Variablen für das Kommando zur Verfügung (z. B. Variable 6 und 7 zur Richtungssteuerung einer RLA4/5 an HF1/VHF und HF2). Die erforderlichen Daten sind den Beschreibungen der Antennen bzw. der Steuergeräte RSW2, 3 oder 4 zu entnehmen.

Hinweis: Bei Aktivierung der Steuergeräte muss die Versorgungsspannung des RSR200 mindestens 0,5 V über der Fernspeisespannung liegen. Der Strom zur Versorgung der Antenne(n) wird über Analog-Spannungsregler direkt der Versorgungsspannung entnommen. Er führt zur zusätzlichen Erwärmung des RSR200 entsprechend der generierten Verlustleistung! Die Versorgungsspannung des RSR200 sollte deshalb nicht viel höher als die Fernspeisespannung + 0,5 V sein. Sie sollte gut stabilisiert und rausch- / störarm sein.

#### 4.5 Verwendung des GPS-Empfängers

Im RSR200 ist ein GPS-Empfänger integriert, der ein 1pps Signal (1 Puls pro Sekunde) ausgibt. Die Genauigkeit des Impulses wird durch die Atomuhren der empfangenen Satelliten und die inherenten Fehlerquellen der Datenübertragung und der Signalverarbeitung im Empfänger bestimmt. Sie ist für den verwendeten Empfänger mit besser 15 ppm angegeben, liegt bei gutem Empfang jedoch typischerweise weit unter 1 ppm. Damit ist dieses Signal als Zeitbasis zur genauen Messung und eventuellen Korrektur der ADC-Taktfrequenz geeignet.

Weitere Daten wie Positionsbestimmungen o. ä. des GPS-Empfängers werden nicht verarbeitet.

Die ADC-Taktfrequenz des RSR200 ist variabel (Kommando „ADC-Takt Einstellen“). Der verwendete Taktoszillator wird dazu mit einem Digitalwort gesteuert. Dieses Datenwort kann benutzt werden, um Abweichungen der Oszillatorfrequenz von der eingestellten Sollfrequenz auszugleichen.

Dazu misst der RSR200 die Oszillatorfrequenz mit dem 1 pps Signal als Zeitbasis in 2 verschiedenen Intervallen:

- 2 Sekunden-Intervall (zwei 1pps Impulse): Auflösung 0,5 Hz.
- 10 Sekunden-Intervall (zehn 1pps Impulse): Auflösung 0,1 Hz.

Über Kommando „ADC-Takt Einstellen“ (Parameter GPS-Dis) kann der RSR200 auf 2 verschiedene Betriebsarten zur Verwendung dieser Messdaten geschaltet werden.

- Interne Regelung: Die Messdaten werden zur Berechnung eines Korrekturwertes verwendet, mit dem der Taktoszillator um genau diesen Wert nachgestimmt wird, so dass er auf der Sollfrequenz arbeitet. Der Korrekturwert repräsentiert also die aktuelle Abweichung des freilaufenden Oszillators ohne Korrektur. Er wird in den Datenblöcken im 2 Sekunden Intervall (0,5 Hz Auflösung) ausgegeben.
- Keine Regelung: Der Oszillator läuft frei, ohne selbständig vom RSR200 korrigiert zu werden. Die Messung der aktuellen Abweichung (Istwert minus Sollwert) wird in den Datenblöcken im 10 Sekunden Intervall (0,1 Hz Auflösung) ausgegeben.

Der RSR200 akzeptiert jederzeit Korrekturdaten für den Taktoszillator mittels Kommando „Variablen 16 Bit-Wert Einstellen“ (Variable Nr. 0) in 0,1 Hz Auflösung. Zusammen mit der Auswertung den vom RSR200 gesendeten Messdaten kann damit eine externe (vom PC geführte) Regelung aufgebaut werden.

Hinweis: Die Nachregelung des Taktoszillators mittels digitalem Steuerwort verursacht bei jeder neuen Einstellung einen Frequenzsprung entsprechend dem Unterschied der bisherigen Einstellung zur neuen Einstellung. Dieser Sprung kann im Signal bemerkbar sein und stören. Nachregelungen des Taktoszillators sollten deshalb immer nur in sehr kleinen Schritten erfolgen (außer eine große Abweichung, z. B. nach neuer Frequenzvorgabe, soll schnell korrigiert werden).

## 4.6 Zweikanalbetrieb

Im Modus „Separate“ ist ein vollkommen unabhängiger Betrieb (verschiedene Eingangssignale und verschiedene Mischfrequenzen) beider Kanäle des RSR200 möglich. Die Kanäle können ab Firmware V223 auch zur vollsynchronen (phasen-korrelierten) Verarbeitung der Eingangssignale bei gleicher Mischfrequenz eingesetzt werden. Dazu wurden die Latenzen (Signallaufzeiten) beider Kanäle exakt angeglichen. Zusätzlich sind Synchronisierungsschaltungen vorhanden, die bei bestimmten Ereignissen die Mischoszillatoren auf gleiche Phasenlage schalten (bei LO2 eventuell unter Berücksichtigung der zusätzlich einstellbaren Phase), sowie alle Ablaufsteuerungen in den Dezimierern synchronisieren. Daraus resultiert eine exakt gleiche Phasenlage (+ Phase LO2) der Signale in beiden Kanälen.

Die Synchronisation erfolgt bei folgenden Ereignissen:

- Änderung der Einstellung des ADC-Taktes
- Änderung der Einstellung der Datenübertragung (Schnittstellen)

Um die phasensynchrone Arbeitsweise der Kanäle nutzen zu können, muss die Sequenz der Kommandos so gestaltet werden, dass nach einem Kommando mit Auslösung des Synchronisations-Ereignisses keine Kommandos folgen, welche die Phasengleichheit wieder zerstören könnten. Beispielsweise ist nach Änderung einer LO-Frequenz (→ unterschiedliche Frequenzen des Signals in den Kanälen) die Phasenlage unbestimmt, wenn später wieder auf gleiche LO-Frequenzen geschaltet wird. Es muss nachfolgend ein Kommando mit Auslösung der Synchronisation gesendet werden. Die gleichzeitige Verstellung beider LO-Frequenzen mit einem Kommando garantiert die Synchronizität, es ist keine erneute Synchronisierung erforderlich.