4.2.16 Miscellaneous Blocks

ReceiverSetup	Number:	5902
	"OnChange" interval:	Block generated each time a user-command is entered to change one or more values in the block (e.g. when entering the **setMarkerParameters** command)

The ReceiverSetup block contains parameters related to the receiver and its installation. When generating RINEX files, this block defines the RINEX file name and the contents of the header.

For all fields containing a string, if the length of the string is lower than the size of the corresponding field, the unused bytes are set to zero.

	Parameter	Type	Units	Do-Not-Use	Description
	`Sync1`	c1			Block Header, see 4.1.1
	`Sync2`	c1			
	`CRC`	u2			
	`ID`	u2			
	`Length`	u2	1 byte		
	`TOW`	u4	0.001 s	4294967295	Receiver time stamp, see 4.1.3
	`WNc`	u2	1 week	65535	
	`Reserved`	u1[2]			2 bytes reserved for future use, to be ignored by decoding software
	`MarkerName`	c1[60]			Marker name (set with **`setMarkerParameters`**).
	`MarkerNumber`	c1[20]			Marker number (set with **`setMarkerParameters`**).
	`Observer`	c1[20]			Observer name (set with **`setObserverParameters`**).
	`Agency`	c1[40]			Observer agency (set with **`setObserverParameters`**).
	`RxSerialNumber`	c1[20]			Receiver serial number.
	`RxName`	c1[20]			Receiver GNSS engine name.
	`RxVersion`	c1[20]			Receiver firmware version.
	`AntSerialNbr`	c1[20]			Serial number of the main antenna (set with **`setAntennaOffset`**).
	`AntType`	c1[20]			Type of the main antenna (set with **`setAntennaOffset`**).
	`deltaH`	f4	1 m		δH offset of the main antenna (set with **`setAntennaOffset`**).
	`deltaE`	f4	1 m		δE offset of the main antenna (set with **`setAntennaOffset`**).
	`deltaN`	f4	1 m		δN offset of the main antenna (set with **`setAntennaOffset`**).
Rev 1	`MarkerType`	c1[20]			Marker type (set with the **`setMarkerParameters`** command).
Rev 2	`GNSSFWVersion`	c1[40]			Version the firmware installed on the receiver.
Rev 3	`ProductName`	c1[40]			Product name.
Rev 4	`Latitude`	f8	1 rad	$-2 \cdot 10^{10}$	Latitude of the reference position, from $-\pi/2$ to $+\pi/2$, positive North of Equator. Use the **`setPVTMode`** command to set the reference position.
	`Longitude`	f8	1 rad	$-2 \cdot 10^{10}$	Longitude of the reference position, from $-\pi$ to $+\pi$, positive East of Greenwich. Use the **`setPVTMode`** command to set the reference position.
	`Height`	f4	1 m	$-2 \cdot 10^{10}$	Ellipsoidal height of the reference position (with respect to WGS84 ellipsoid). Use the **`setPVTMode`** command to set the reference position.
	`StationCode`	c1[10]			Station code (set with **`setMarkerParameters`**). This field can for example contains the four-letter IGS station code assigned to the receiver.
	`MonumentIdx`	u1			Monument index (set with **`setMarkerParameters`**). This index is used to identify the monument when there are multiple monuments at the same station.
	`ReceiverIdx`	u1			Receiver index (set with **`setMarkerParameters`**). This index is used to identify the receiver when there are multiple receivers at the same monument.
	`CountryCode`	c1[3]			ISO 3-character country code (set with the **`setMarkerParameters`** command).
	`Reserved1`	c1[21]			Reserved.
	`Padding`	u1[..]			Padding bytes, see 4.1.5

RxMessage	Number:	4103
	"OnChange" interval:	block generated each time a message needs to be sent

The receiver generates ASCII messages to help users follow the progress of processes such as file logging or FTP push (activity log). These messages are output in the RxMessage block, and they can also be retrieved from the command line using the lif,RxMessages command.

Parameter	Type	Units	Do-Not-Use	Description
Sync1	c1			Block Header, see 4.1.1
Sync2	c1			
CRC	u2			
ID	u2			
Length	u2	1 byte		
TOW	u4	0.001 s	4294967295	Receiver time stamp, see 4.1.3
WNc	u2	1 week	65535	
Type	u1		255	Type of message contained in this block: 1: Asynchronous command reply 2: Message about internal logging 3: Message about FTP push 4: Message about Receiver Status 5: Message from slave GNSS receiver 6: Message about CloudIt
Severity	u1		255	Message severity: 1: Info 2: Warning 3: Error
MessageID	u4		0	A unique value associated to each message. This is a counter starting at 1 for the first message after boot and incrementing at each message.
StringLn	u2			Length of Message in characters, including the terminating \0.
Reserved2	u1[2]			Reserved, contents to be ignored.
Message	c1[StringLn]			Receiver message terminated by \0.
Padding	u1[..]			Padding bytes, see 4.1.5

Commands	Number: 4015 "OnChange" interval: each time a user command is entered

Every time the user sends a command, a Commands block is output on all ports for which this block is enabled. The Commands SBF block is inserted in the SBF stream at the very moment when the command starts to take effect.

Parameter	Type	Units	Do-Not-Use	Description
Sync1	c1			Block Header, see 4.1.1
Sync2	c1			
CRC	u2			
ID	u2			
Length	u2	1 byte		
TOW	u4	0.001 s	4294967295	Receiver time stamp, see 4.1.3
WNc	u2	1 week	65535	
Reserved	u1[2]			Reserved for future use, to be ignored by decoding software.
CmdData	u1[*N*]			Command data, this is the command in the SNMP' format (reserved for maintenance and support only).
Padding	u1[..]			Padding bytes, see 4.1.5

Comment	Number:	5936
	"OnChange" interval:	block generated each time a comment is entered with **setObserverComment**

The Comment block contains a comment string as entered with the **setObserverComment** command.

Parameter	Type	Units	Do-Not-Use	Description
Sync1	c1			Block Header, see 4.1.1
Sync2	c1			
CRC	u2			
ID	u2			
Length	u2	1 byte		
TOW	u4	0.001 s	4294967295	Receiver time stamp, see 4.1.3
WNc	u2	1 week	65535	
CommentLn	u2			Length of the Comment string, in characters. The maximum length of a comment is 120 characters.
Comment	c1[CommentLn]			Comment string, as entered with the **setObserverComment** command. Note that this string is not terminated by the "\0" character.
Padding	u1[..]			Padding bytes, see 4.1.5

BBSamples	Number:	4040
	"OnChange" interval:	block generated each time new baseband samples are ready (typically at 2Hz)

The BBSamples block contains a series of successive complex baseband samples. These samples can be used for signal monitoring and for spectral analysis of the GNSS bands supported by the receiver.

Parameter	Type	Units	Do-Not-Use	Description
Sync1	c1			Block Header, see 4.1.1
Sync2	c1			
CRC	u2			
ID	u2			
Length	u2	1 byte		
TOW	u4	0.001 s	4294967295	External time stamp, see 4.1.3
WNc	u2	1 week	65535	
N	u2			Number of complex baseband samples contained in this block
Info	u1			Bit field as follows: Bits 0-2: Antenna ID: antenna from which the samples have been taken: 0 for main, 1 for *Aux1* and 2 for *Aux2*. Bits 3-7: Reserved
Reserved	u1[3]			Reserved for future use, to be ignored by decoding software.
SampleFreq	u4	1 Hz		Sampling frequency in Hz.
LOFreq	u4	1 Hz		Frequency of the local oscillator (LO) used to down-convert the RF signal to baseband.
Samples	u2[N]			N successive complex baseband samples (I+jQ), coded as follows: Bits 0-7: 8-bit Q component, two's complement. Bits 8-15: 8-bit I component, two's complement.
Padding	u1[..]			Padding bytes, see 4.1.5

ASCIIIn	Number:	4075
	"OnChange"	interval: block generated each time an ASCII string is received

The ASCIIIn block contains a string that has been received on one of the receiver's connection ports.

More specifically, this block is output each time an end-of-line character is received on a communication port configured to receive ASCIIIn input (with the **setDataInOut** command). The string reported in this block contains all characters received since the previous occurrence of an end-of-line character.

The maximum length of the string is 2000 characters. If there are more than 2000 characters between the occurrence of two successive end-of-line characters, the string is discarded

<table>
<tr><th>Parameter</th><th>Type</th><th>Units</th><th>Do-Not-Use</th><th>Description</th></tr>
<tr><td>Sync1</td><td>c1</td><td></td><td></td><td rowspan="5">Block Header, see 4.1.1</td></tr>
<tr><td>Sync2</td><td>c1</td><td></td><td></td></tr>
<tr><td>CRC</td><td>u2</td><td></td><td></td></tr>
<tr><td>ID</td><td>u2</td><td></td><td></td></tr>
<tr><td>Length</td><td>u2</td><td>1 byte</td><td></td></tr>
<tr><td>TOW</td><td>u4</td><td>0.001 s</td><td>4294967295</td><td rowspan="2">Receiver time stamp, see 4.1.3</td></tr>
<tr><td>WNc</td><td>u2</td><td>1 week</td><td>65535</td></tr>
<tr><td>CD</td><td>u1</td><td></td><td></td><td>Identifier of the connection from which the data has been received:
<table>
<tr><th>Value of CD</th><th>Connection type</th><th>Example</th></tr>
<tr><td>0-31</td><td>COMx, with <i>x</i>=CD</td><td>1: COM1</td></tr>
<tr><td>32-47</td><td>USBx, with <i>x</i>=CD-32</td><td>33: USB1</td></tr>
<tr><td>48-63</td><td>OTGx, with <i>x</i>=CD-48</td><td>49: OTG1</td></tr>
<tr><td>64-95</td><td>IPx, with <i>x</i>=CD-54</td><td>64:IP10</td></tr>
<tr><td>128-159</td><td>NTRx, with <i>x</i>=CD-128 (NTRIP connections)</td><td>129:NTR1</td></tr>
<tr><td>192</td><td>BT01 (Bluetooth connection)</td><td></td></tr>
<tr><td>193</td><td>BT02 (Bluetooth connection)</td><td></td></tr>
<tr><td>196</td><td>UHF1 (UHF Modem)</td><td></td></tr>
<tr><td>200-205</td><td>IPRx, with <i>x</i>=CD-200 (IP receive connections)</td><td>201:IPR1</td></tr>
<tr><td>210</td><td>DCL1 (cellular data-call connection)</td><td></td></tr>
<tr><td>214</td><td>CAN1 (CAN stream interface)</td><td></td></tr>
<tr><td>215-255</td><td>Reserved</td><td></td></tr>
</table></td></tr>
<tr><td>Reserved1</td><td>u1[3]</td><td></td><td></td><td>Reserved, contents to be ignored.</td></tr>
<tr><td>StringLn</td><td>u2</td><td></td><td></td><td>Length of ASCIIString in characters.</td></tr>
<tr><td>SensorModel</td><td>c1[20]</td><td></td><td></td><td>Not supported, reserved for future use.</td></tr>
<tr><td>SensorType</td><td>c1[20]</td><td></td><td></td><td>Not supported, reserved for future use.</td></tr>
<tr><td>Reserved2</td><td>u1[20]</td><td></td><td></td><td>Reserved, contents to be ignored.</td></tr>
<tr><td>ASCIIString</td><td>c1[StringLn]</td><td></td><td></td><td>ASCII string. Note that this string is not terminated by the "\0" character. The string does not include the end-of-line character(s) (carrier return and/or line feed).</td></tr>
<tr><td>Padding</td><td>u1[..]</td><td></td><td></td><td>Padding bytes, see 4.1.5</td></tr>
</table>

EncapsulatedOutput	Number:	4097
	"OnChange" interval:	output each time an RTCM, CMR, NMEA or ASCIIDisplay message is output

The EncapsulatedOutput block encapsulates non-SBF output messages into SBF. It is enabled with the Encapsulate option of the **setDataInOut** command.

Parameter	Type	Units	Do-Not-Use	Description
Sync1	c1			Block Header, see 4.1.1
Sync2	c1			
CRC	u2			
ID	u2			
Length	u2	1 byte		
TOW	u4	0.001 s	4294967295	Receiver time stamp, see 4.1.3
WNc	u2	1 week	65535	
Mode	u1			Type of the message encapsulated in the Payload field: 0: RTCMv2 1: CMRv2 2: RTCMv3 4: NMEA 5: ASCIIDisplay
Reserved	u1			Reserved for future use, to be ignored by decoding software.
N	u2			Length of Payload in bytes.
ReservedId	u2			Reserved for future use
Payload	u1[N]			Encapsulated message.
Padding	u1[..]			Padding bytes, see 4.1.5