uAvionix skyLink Interface Control Document

UAV-1004282-001

Rev G

May 11, 2022

300 Pine Needle Lane

Bigfork, MT 59911

http://www.uavionix.com

support@uavionix.com

Revision	Date	Comments
A	7/02/2019	Initial release.
B	6/25/2020	Add Masked Hop Table Configuration Data messages. Add Transmit-Receive Mask messages. Added STANDBY/ACTIVE/OVERRIDE modes. Formatting and section numbering.
C	10/06/2020	Add NMEA input support via Control port Removed electrical interface specifications
D	1/26/2021	Add UCP Identification message Add firmware revision and ICD correlation table Add Advanced Configuration message Add clarifying descriptions of subfield in Status message.
E	2/12/2021	Bitfield location corrections in Configuration Data message and Hop Table ID in all Hop Table messages Add Message Request Message
F	4/28/2021	Add Advanced Status message
G	5/11/2022	V2 and V3 Advanced Status Message

Contents

1 Async HDLC

The datalink message structure is based on “Async HDLC”, as described in Minimum Aviation System Performance Standards (MASPS) for Flight Information Services-Broadcast (FIS-B) Data Link (RTCA/DO-267), Section 3.4.3.2.

2 Document Scope

This document describes the control and configuration messages used by the skyLink system’s serial and ethernet CTRL interfaces. This document does not describe the details of the over-the-air RF data link.

3 Physical Interfaces

In general, messages defined in this document may be exchanged over the CNTL serial link or network connection.

3.1 Serial Interface

The Airborne Radio System (ARS) has three physical serial data interfaces and one Pulse-Per-Second input signal. The interfaces are Control (CTRL), User (USER) and GPS + 1 PPS.

Communication to an ARS device is accomplished via the CNTL interface through a full duplex asynchronous serial connection. Depending on the device, this can be TTL or RS-232. The interface is configured with the following characteristics:

Default Data bit rate:	115200bps
Data Length:	8 bits
Parity:	None
Stop Bits:	1 bit
Flow Control:	None

3.2 Network Interface

The Ground Radio System (GRS) has an ethernet interface that support CTRL data via standard Internet Protocols and transports.

4 Message Format

The uAvionix CTRL messages follow the Async HDLC for framing, byte stuffing, control escape, and frame check sequencing. The standard flag byte of 0x7E and escape byte of 0x7D are used. Any data field not specified as “MSB first” is transmitted in little-endian order.

Frame Flag (0x7E)	Msg ID 1 byte	Payload N bytes	Frame Check Sequence 2 bytes	Frame Flag (0x7E)

4.1 Message Frame Check Sequence

The frame check sequence is a CRC-CCITT. A table generated CRC calculation may be used. This table contains 256 elements and should be calculated at startup and left unchanged afterward. An example method for table construction is provided.

```
Void crcInit( void )
{
        unsigned int i, bitctr, crc;
        for (i = 0; i < 256; i++)
        {
                crc = (i << 8);
                for (bitctr = 0; bitctr < 8; bitctr++)
                {
                        crc = (crc << 1) ^ ((crc & 0x8000) ? 0x1021 : 0);
                }
                Crc16Table[i] = crc;
        }
}
```

The CRC calculation can be produced with the following method.

```
Unsigned int crcCompute(              // Return – CRC of the block
      unsigned char *block,           // i – Starting address of message
      unsigned long int length,       // i – Length of message
      )
{
      unsigned long int i;
      unsigned int crc = 0;
      for (i = 0; i < length; i++)
      {
               crc = Crc16Table[crc >> 8] ^ (crc << 8) ^ block[i];
      }
      return crc;
}
```

4.2 UCP Messages

4.2.1 UCP Message Types

UCP Message Types	Version	I/O	Section
44_{10} (0x2C) – Message Request	2	In	4.2.1.1
37_{10} (0x25) – Identification Message	3	Out	4.2.1.2

4.2.1.1 44_{10} – Message Request Message

The Message Request Message is used to elicit a message from the target device.

Offset	Type	Value	Description
0	uint8_t	0x2C	Message ID
1	uint8_t	0x02	Protocol Version

2	uint8_t	0x25	Requested Message ID

4.2.1.2 37_{10} – Identification Message

If the device is queried with a UCP Identification message request, this message will be output containing the current information about the device and will serve as an identification of the device.

Offset	Type	Value	Description
0	uint8_t	0x25	Message ID
1	uint8_t	0x03	Protocol Version
2	Device ID		Primary Device ID. See section 4.2.1.3
14	Device ID		Secondary Device ID. See section 4.2.1.3
26	Firmware ID		Primary Firmware ID. See section 4.2.1.4
31	Firmware ID		Secondary Firmware ID. See section 4.2.1.4
36	uint8_t[15]		Primary Firmware Part Number (not null terminated, but null padded if part number is less than 15 characters)
51	uint8_t[15]		Secondary Firmware Part Number (not null terminated, but null padded if part number is less than 15 characters)

4.2.1.3 UCP Device ID

Contains the firmware version, hardware ID and serial number of a device.

Set to all 0xFF if invalid.

Offset	Type	Value	Description
0	uint8_t		Major FW Version Number
1	uint8_t		Minor FW Version Number
2	uint8_t		Build FW Version Number
3	uint8_t		HW ID
4	uint64_t		Radio ID/Serial Number of the device

4.2.1.4 GDL90 Firmware ID

Offset	Type	Value	Description

0	uint8_t		Firmware Identification
1	uint32_t		Firmware CRC

4.3 skyLink Messages

Most messages pertaining to the control and configuration of skyLink will contain a Message ID of 49_{10} (0x31) and shall follow the format below. The Message ID SubType will further define the message format.

Frame Flag (0x7E)	Msg ID: 49_{10} (0x31)	Msg ID SubType: 1 byte	Payload: N bytes	Frame Check Sequence: 2 bytes	Frame Flag (0x7E)

uAvionix skyLink supported messages are broken down into two categories related to the purpose of the message. The first category is Configuration and relates to the initial setup of the skyLink device to the system or aircraft it's being installed into. The second category is Operational, which relates to the ongoing operation of the skyLink device once it has been installed into the aircraft and been configured.

4.3.1 Configuration Messages

Message ID SubType & Name	I/O	Section
02_{10} (0x02) – Configuration Request	In	4.3.1.1
03_{10} (0x03) – Configuration Data	In/Out	4.3.1.2
04_{10} (0x04) – Hop Table Request	In/Out	4.3.1.3
05_{10} (0x05) – Hop Table Configuration Data	In/Out	4.3.1.4
06_{10} (0x06) – Reserved		
07_{10} (0x07) – Reserved		
08_{10} (0x08) – Masked Hop Table Request	In	4.3.1.5
09_{10} (0x09) – Masked Hop Table Configuration Data	In/Out	4.3.1.6
10_{10} (0x0A) – Transmit-Receive Mask Request	In	4.3.1.7
11_{10} (0x0B) – Transmit-Receive Mask Configuration Data	In/Out	4.3.1.8

12_{10} (0x0C) – Reserved		
13_{10} (0x0D) – Reserved		
14_{10} (0x0E) – Reserved		
15_{10} (0x0F) – Advanced Configuration Request	In	4.3.1.9
16_{10} (0x10) – Advanced Configuration Data	In/Out	4.3.1.10
17_{10} (0x11) – Reserved		
18_{10} (0x12) – Reserved		

4.3.1.1 Configuration Request (Msg ID SubType = 02_{10})

When a Configuration Request message is sent to the skyLink device, the device will reply with a Configuration Data (Msg ID SubType = 03_{10}) message in response. This allows a method for the host to retrieve the current configuration for initialization and verification purposes.

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x02	skyLink Message ID SubType

4.3.1.2 Configuration Data (Msg ID SubType = 03_{10})

The Configuration Data message should be sent only as required to set or update the device's version of the information fields contained within this message. Valid receipt of this message will cause the device to write the new data into non-volatile memory. All data will persist through a power cycle. If a confirmation of reception is desired, simply send a Configuration Request (Msg ID SubType = 02_{10}) message after sending the configuration message and verify field data match.

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x03	skyLink Message ID SubType
2	uint8_t		Station Type (4.3.3.3)
3	uint8_t		UTC Pulse Polarity (4.3.3.4)
4	uint32_t		User Port Baud Rate in bps
8	uint32_t		Position Port Baud Rate in bps

12	uint32_t		Control Port Baud Rate in bps
16	uint8_t(0:0)		UART Idle Framing Enabled
16	uint8_t(1:1)		Stale Data Framing Enabled
16	uint8_t (7:2)	0	Spare Bits (Set to 0's)
17	uint8_t		Maximum Transmission Unit (MTU) in bytes

4.3.1.3 Hop Table Request (Msg ID SubType = 03_{10})

When a Hop Table Request message is sent to the skyLink device, the device will reply with a Hop Table Configuration Data (Msg ID SubType = 04_{10}) message containing the requested section of the hop table. This allows a method for the host to retrieve a section of a hop table for initialization and verification purposes. A Hop Table Request message can also be sent by the device to request additional hop channels as part of an entire hop table transfer. If a Hop Table Request message containing a "First Hop Channel Requested" greater than the size of the Hop Table, the message will be considered invalid and ignored.

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x04	skyLink Message ID SubType
2	uint8_t(1:0)		Hop Table ID (RFU, set to 0)
2	uint8_t(7:2)		Spare bits (Set to 0's)
3	uint8_t		First Hop Channel Requested

4.3.1.4 Hop Table Configuration Data (Msg ID SubType = 04_{10})

The Hop Table Configuration Data message should be sent only as required to set or update the device's version of the Hop Table contained within this message. If a confirmation of reception is desired, the host should simply send a Hop Table Request (Msg ID SubType = 0310) message after sending the Hop Table Configuration Data message and verify the data matches. The maximum number of Hop Channels in any Hop Table Configuration Data message is 25 which maintains a maximum message size of 206 bytes. Hop Tables are stored in non-volatile memory and will be retained through a power cycle. Only one Hop Table can be stored at a time. All non-volatile transmit enable and receive enable masks will be set to true and any masks set by Masked Hop Table Configuration Data (Msg ID SubType = 09_{10}) will be erased.

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID

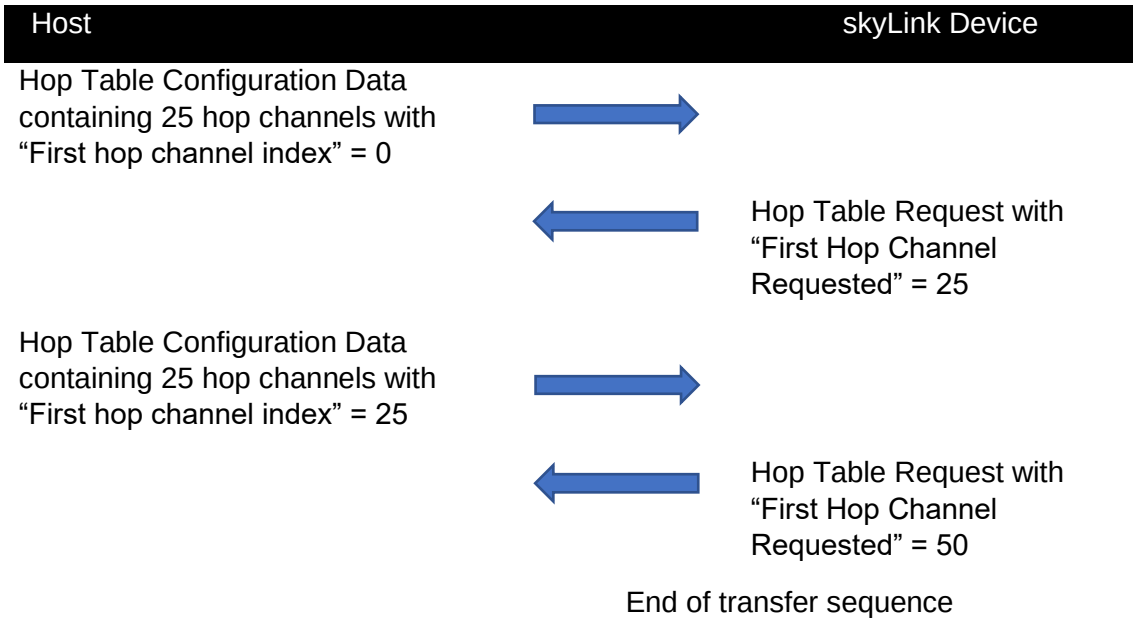
1	uint8_t	0x05	skyLink Message ID SubType
2	uint8_t(1:0)		Hop Table ID (RFU, set to 0)
2	uint8_t(7:2)		Spare bits (Set to 0's)
3	uint8_t		Total number of Hop Channels in table (note: multiple Hop Channel Configuration Data messages are required to specify a complete table)
4	uint8_t		Number of Hop Channels in this message
5	uint8_t		First Hop Channel Index. A 0-based index of first Hop Channel specified in this message
6	Hop Channel		HopChannel[First Hop Channel Index + 0] (4.3.1.4.1)
14	Hop Channel		HopChannel[First Hop Channel Index + 1]
…			…
198	Hop Channel		HopChannel[First Hop Channel Index + 24]

4.3.1.4.1 Hop Channel

Offset	Type/Width	Description
0	uint32_t	Frequency in Hz
4	uint32_t	Channel Code Word

4.3.1.4.2 Hop Table Transfer Sequence Example

This is an example sequence of how the Hop Table Request (Msg ID SubType = 0310) and Hop Table Configuration Data (Msg ID SubType = 0410) messages are used to transfer a new hop table to a skyLink device.



4.3.1.5 Masked Hop Table Request (Msg ID SubType = 08_{10})

When a Masked Hop Table Request message is sent to the skyLink device, the device will reply with a Masked Hop Table Configuration Data (Msg ID SubType = 09_{10}) message containing the requested section of the hop table. This allows a method for the host to retrieve a section of a hop table for initialization and verification purposes. A Masked Hop Table Request message can also be sent by the device to request additional hop channels as part of an entire hop table transfer. If a Masked Hop Table Request message containing a "First Hop Channel Requested" greater than the number of entries in the Hop Table, the message will be considered invalid and ignored.

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x08	skyLink Message SubType ID
2	uint8_t	0x01	Message format version number
3	uint8_t(1:0)		Hop Table ID (RFU, set to 0)
3	uint8_t(7:2)		Spare bits (Set to 0's)
4	uint8_t		0-based index of first Masked Hop Channel Requested

4.3.1.6 Masked Hop Table Configuration Data (Msg ID SubType = 09_{10})

The Masked Hop Table Configuration Data message is used to selectively enable and disable transmit and receive operations on a per hop basis. Transmit and receive can be

controlled independently for each hop entry. The Masked Hop Table will be committed to non-volatile memory and will be persistent across power cycles.

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x09	skyLink Message ID SubType
2	uint8_t	0x01	Message format version number
3	uint8_t(1:0)		Hop Table ID (RFU, set to 0)
3	uint8_t(7:2)		Spare bits (Set to 0's)
4	uint8_t		Total number of Hop Channels in table (note: multiple Masked Hop Channel Configuration Data messages are required to specify a complete table)
5	uint8_t		Number of Hop Channels in this message
6	uint8_t		First Hop Channel Index. A 0-based index of first Hop Channel specified in this message
7	Masked Hop Channel		Channel[First Hop Channel Index + 0]
16	Masked Hop Channel		Channel[First Hop Channel Index + 1]
…			…
196	Masked Hop Channel		Channel[First Hop Channel Index + 20]

The Masked Hop Table Configuration Data message should be sent only as required to set or update the device's version of the Masked Hop Table contained within this message. If a confirmation of reception is desired, the host should simply send an Masked Hop Table Request message after sending the Masked Hop Table Configuration message and verify the data matches. The maximum number of Masked Hop Channels in any Masked Hop Table Configuration Data message is 21 to maintain a maximum message size less than 204 bytes.

4.3.1.6.1 Masked Hop Channel

Offset	Type/Width	Description
0	uint32_t	Frequency (Hz)

4	uint32_t	Channel Code Word
8	uint8_t(7:5)	RFU, set to 0
8	uint8_t(4:4)	1 = Transmit enabled, 0 = Transmit disabled
8	uint8_t(3:1)	RFU, set to 0
8	uint8_t(0:0)	1 = Receive enabled, 0 = Receive disabled

4.3.1.7 Transmit-Receive Mask Request (Msg ID SubType = 10_{10})

The Transmit-Receive Mask Request message is used to request a Transmit-Receive Mask Configuration Data (Msg ID SubType = 11_{10}) message.

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x0A	skyLink Message ID SubType
2	uint8_t	0x01	Message format version number

4.3.1.8 Transmit-Receive Mask Configuration Data (Msg ID SubType = 11_{10})

The Transmit-Receive Mask Configuration Data message is used to control transitions from ACTIVE (transmitting on some or all Hop Channels) to STANDBY (non-transmitting). To command a STANDBY state, set the Mode subfield to 0. If the Mode subfield is 0, the Hop Channel Transmit enable/disable and the Hop Receive enable/disable mask subfields are ignored. To command an ACTIVE state, set the Mode subfield to either 1 or 2. If the Mode subfield is set to 1, the Hop Channel Transmit enable/disable and the Hop Receive enable/disable mask subfields are ignored and the non-volatile Hop Channel masks are used (as specified in Masked Hop Table Configuration Data (Msg ID SubType = 09_{10}). If the Mode subfield is set to 2, the Hop Transmit enable/disable and the Hop Receive enable/disable mask subfields from this message are used instead of masks set using Masked Hop Table Configuration Data messages. The masks specified in this message are not persistent and will be discarded on a power cycle at which point the Masked Hop Table Configuration Data masks will be restored.

4.3.1.8.1 Version 1

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x0B	skyLink Message ID SubType
2	uint8_t	0x01	Message format version number

3	uint8_t		**Mode.** 0 = STANDBY, 1 = Use non-volatile hop table masks, 2 = Masks in this message override the non-volatile hop table masks. If Mode = 2, the currently active non-volatile mask in use will be returned. Note that 0 and 2 will enable the age out of items in the transmit queue unless explicitly disabled in the transmit queue age out field of the Advanced Configuration Data message
4	uint8_t[8]		Hop Channel Transmit enable/disable bitmask. Mask[0](7:7) – HopTable[0] Mask[0](6:6) – HopTable[1] Mask[0](5:5) – HopTable[2] … Mask[6](7:7) – HopTable[48] Mask[6](6:6) – HopTable[49] Mask[6](5:0) – Reserved, set to 0 Mask[7](7:0) - Reserved, set to 0
12	uint8_t[8]		Hop Channel Receive enable/disable bitmask. Mask[0](7:7) – HopTable[0] Mask[0](6:6) – HopTable[1] Mask[0](5:5) – HopTable[2] … Mask[6](7:7) – HopTable[48] Mask[6](6:6) – HopTable[49] Mask[6](5:0) – Reserved, set to 0 Mask[7](7:0) - Reserved, set to 0

4.3.1.9 Advanced Configuration Request

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x0F	skyLink Message SubType ID
2	uint8_t	0	Message version format

4.3.1.10 Advanced Configuration Data

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x10	skyLink Message SubType ID
2	uint8_t	0x01	Message version format
3	uint8_t[5]		Local node ID (the ID of the node that generated this response). Set to 0 for “do not modify”
8	uint8_t[5]		Remote node ID (the ID of the node that is RF linked to the node that generated this response) (Read only field)
13	uint8_t(7:0)		Reserved for future use. Set to 0.
14	uint16_t		Transmit Queue age out (milliseconds) 0 = do not discard entries (default) 0xFFFF = do not modify existing setting/use the default setting driven by ‘mode’ value in the TxRx Mask Configuration Data message Values 1-99 result in 100 ms age out value All other nonzero values = age in milliseconds after which to discard transmit queue entries not yet sent. (rounded up to the nearest 50 ms multiple)

4.3.2 Operational Messages

Message ID & Name	I/O	Section
01_{10} (0x01) – Status Message	Out	4.3.2.1
19_{10} (0x13) – Advanced Status Message	Out	4.3.2.2

4.3.2.1 Status Message (Msg ID SubType = 01_{10})

The Status message contains all data to represent the immediate status of the local and remote skyLink. This status message is sent out the CTRL serial port of both the ARS and the GRS. The local and remote designators are relative to the originator of the message. Specifically, when the ARS sends this message, the Local skyLink Status subfield refers to the RF receivers of the ARS. The Remote skyLink Status subfield refers to the RF receivers of the GRS. When the GRS sends this message, the Local skyLink Status subfield refers to the RF receivers of the GRS and the Remote skyLink Status subfield refers to the RF receivers of the ARS. Note that the data regarding the Remote receiver is exchanged between the two radios as part of an over-the-air link management protocol. That data is made available to the user on the ground and to the aircraft flight control module via this message over the CTRL port. Status messages are sent twice per second when the skyLink is operational with valid position data and UTC pulses.

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x01	skyLink Message ID SubType
2	skyLink Status		Local skyLink Status (4.3.3.1)
84	skyLink Status		Remote skyLink Status (4.3.3.1)

4.3.2.2 Advanced Status Message (Msg ID SubType = 19_{10})

Similar to the Status message (0x01) but with a version subfield, and Status Data Source records for each of the two devices.

4.3.2.2.1 Version 1

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x13	skyLink Message ID SubType
2	uint8_t	0x01	Message version
3	skyLink Status		Local skyLink Status (4.3.3.2)
85	skyLink Status		Remote skyLink Status (4.3.3.2)

167	skyLink Status Data Source		Local Data Source (4.3.3.1)
180	skyLink Status Data Source		Remote Data Source (4.3.3.1)

4.3.2.2.2 Version 2

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x13	skyLink Message ID SubType
2	uint8_t	0x02	Message version
3	skyLink Status		Local skyLink Status (4.3.3.2)
85	skyLink Status		Remote skyLink Status (4.3.3.2)
167	skyLink Status Data Source		Local Data Source (4.3.3.1)
180	skyLink Status Data Source		Remote Data Source (4.3.3.1)
193	int8_t		Local Radio Board Temperature (°C)
194	int8_t		Remote Radio Board Temperature (°C)

4.3.2.2.3 Version 3

Offset	Type/Width	Value	Description
0	uint8_t	0x31	Message ID
1	uint8_t	0x13	skyLink Message ID SubType
2	uint8_t	0x02	Message version
3	skyLink Status		Local skyLink Status (4.3.3.2)

85	skyLink Status		Remote skyLink Status (4.3.3.2)
167	skyLink Status Data Source		Local Data Source (4.3.3.1)
180	skyLink Status Data Source		Remote Data Source (4.3.3.1)
193	int8_t		Local Radio Board Temperature (°C)
194	int8_t		Remote Radio Board Temperature (°C)
195	uin8_t(0:1)		Local Radio PPS Sync Status (4.3.3.6)
195	uin8_t(2:3)		Remote Radio PPS Sync Status (4.3.3.6)
195	uin8_t(4:6)		Fault Code (4.3.3.7)
195	uin8_t(7)		Remote radio fault – If 1 fault code is reported by the remote radio, otherwise the fault is from the local radio. Remote faults are only reported if there is no local fault.

4.3.3 Message Field Data Definitions

4.3.3.1 skyLink Status Data Source

Offset	Type/Width	Description
0	uint64_t	Message creation time. Milliseconds since GPS Epoch.
8	uint8_t[5]	Radio ID

4.3.3.2 skyLink Status Structure

Offset	Type/Width	Description
0	int32_t	Latitude in Decimal Degree with $1x10^7$ units
4	int32_t	Longitude in Decimal Degree with $1x10^7$ units
8	int32_t	GNSS Altitude (mm)

12	uint32_t	Horizontal Accuracy (mm)
16	uint32_t	Vertical Accuracy (mm)
20	uint32_t	Velocity Accuracy (mm/s)
24	int32_t	Vertical Velocity (mm/s)
28	int32_t	North Velocity (mm/s)
32	int32_t	East Velocity (mm/s)
36	uint8_t	GPS Fix Type (4.3.3.5)
37	uint8_t	Number of GPS Satellites in solution
38	uint16_t	Transmitted Frames – Number of hops where user data was collected in a frame and transmitted
40	uint16_t	Received Frames – Number of hops where user data frames were captured by one or both receivers.
42	uint16_t	Dropped Frames – Number of hops that were designated for reception but neither receiver captured any data.
44	uint16_t(15:2)	Spare bits (Set to 0's)
44	uint16_t(1:1)	Secondary Radio Link Status. 0 = No Link, 1 = Link OK. Link OK is defined as a non-zero number of user data frames or beacons received in the prior second.
44	uint16_t(0:0)	Primary Radio Link Status. 0 = No Link, 1 = Link OK. Link OK is defined as a non-zero number of user data frames or beacons received in the prior second.
46	uint32_t	Transmitted Bytes Total – Total number of user data bytes transmitted since the most recent change of remote node.
50	uint32_t	Received Bytes Total – Total number of user data bytes received since the most recent change of remote node.
54	uint16_t	Transmitted Frame Rate (frames/second) – User data frames/second in the last evaluation interval (1 second)

56	uint16_t	Receive Frame Rate (frames/second) - User data frames/second in the last evaluation interval (1 second)
58	uint16_t	Transmitted Data Rate (bytes/second) - User data bytes/second in the last evaluation interval (1 second)
60	uint16_t	Received Data Rate in (bytes/second) - User data bytes/second in the last evaluation interval (1 second)
62	uint8_t	Receive Frame Queue Depth - % full of the receive queue at the time this message was created
63	uint8_t	Receiver Max Queue Depth – high water mark of the queue depth since the most recent change of remote node.
64	uint8_t	Transmitter Queue Depth - % full of the receive queue at the time this message was created
65	uint8_t	Transmitter Max Queue Depth – high water mark of the queue depth since the most recent change of remote node.
66	int16_t	Primary Receiver: RSSI (dBm x 10^2) – exponential moving average smoothed value of RSSI recorded by the receiver. This value is only valid if the Primary Link Status = 1.
68	uint16_t	Primary Receiver: Received Beacon Frame Count – Number of hops where a beacon or control data was received
70	uint16_t	Primary Receiver: Received Data Frame Count – Number of hops where user data was received
72	uint16_t	Primary Receiver: Dropped Frame Count – number of hops where nothing was received.
74	int16_t	Secondary Receiver RSSI (dBm x 10^2) – exponential moving average smoothed value of RSSI recorded by the receiver. This value is only valid if the Secondary Link Status = 1.
76	uint16_t	Secondary Receiver Received Beacon Frame Count – Number of hops where a beacon or control data was received

78	uint16_t	Secondary Receiver Received Data Frame Count – Number of hops where user data was received
80	uint16_t	Secondary Receiver Dropped Frame Count – number of hops where nothing was received.

4.3.3.3 Station Type

Station Type	Value
Ground	0
Airborne	1
Infrastructure (Used for ground radios connected to SkyLine)	2

4.3.3.4 UTC Pulse Polarity

Pulse Polarity	Value
Pulse Positive (Rising edge Captured)	0
Pulse Negative (Falling edge Captured)	1

4.3.3.5 GPS Fix Type

Fix Type	Value
None	0
No Fix	1
2D Fix	2
3D Fix	3
3D Fix with Differential	4
3D Fix with RTK	5

4.3.3.6 PPS Sync Status

Radio Type	Value
Not synced to PPS	0
Synced to PPS	1

PPS lost, synced to internal timer	2

4.3.3.7 Fault Code

Radio Type	Value
No Fault	0
FPGA Fault	1
RF Fault	2