Using the SM200A with the VITA Radio Transport (VRT) Protocol

User Guide

User’s Guide to Using the Signal Hound SM200A with the ANSI/VITA 49 Radio Transport Protocol

1502 SE Commerce Ave, Suite 101
Battle Ground, WA 9862
Phone 360.217.0112

Contents

1 Overview

This document outlines the process of using the Signal Hound SM200A spectrum analyzer to obtain IQ data contained in packets that conform to the ANSI/VITA 49 Radio Transport (VRT) standard.

The latest version of the standard, 49.2-2017 is used, however the only effective differences insofar as this document and the parts of the protocol used by Signal Hound are concerned are nominal. For example, the IF Data Packet in V49.0 is called the Signal Data Packet in V49.2.

1.1 Introduction

VRT is an open radio transport protocol used to transmit and receive sample data between devices. It is defined by the VITA 49 standard.

1.2 Requirements

The following items are required to use the Signal Hound SM200A to obtain VRT packets:

- The Signal Hound SM200A spectrum analyzer with required USB cable
- A Windows 7/8/10 computer meeting minimum requirements to run Spike software (see Spike Manual for details)

1.3 Installation

1.3.1 Software Installation

1.3.1.1 Windows

- Install Spike to install the SM200A drivers. Download the Signal Hound SDK for access to the SM200A API and VRT examples.

1.3.2 Device Connection

1.3.2.1 Windows

Refer to the Spike Manual to get your Signal Hound device connected and running with your computer. When working properly in Spike you are ready to use the VRT API functions.

2 Specification

2.1 Type and Function of Packets

VRT uses Signal Data packets and Context packets. Both types contain headers with metadata which includes a Stream Identifier and timestamp.

Signal Data packets encapsulate variable-sized blocks of IQ data, along with a 32-bit trailer to convey additional critical information about the state of the receiver at the time the samples were obtained. For example, if the system was being overdriven this would be reported by an indicator in the trailer.

Context packets contain information about the receiver's settings. They are of variable size, depending on how many of the possible ~25 fields are used. Which fields are used is communicated by the Context Indicator field, a 32-bit value which precedes the context fields. Signal Hound uses 10 of the possible context fields.

What follows is a precise specification of how each kind of packet is structured, as defined by the VITA 49 standard[1].

2.2 Signal Data Packet

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="4">Pkt Type</td><td>C</td><td>T</td><td>N</td><td>S</td><td colspan="2">TSI</td><td colspan="2">TSF</td><td colspan="4">Pkt Count</td><td colspan="16">Packet Size</td></tr>
<tr><td colspan="32">Stream Identifier</td></tr>
<tr><td colspan="32">Integer Timestamp - Seconds</td></tr>
<tr><td colspan="32">Fractional Timestamp - Picoseconds (Most significant word)</td></tr>
<tr><td colspan="32">Fractional Timestamp - Picoseconds (Least significant word)</td></tr>
<tr><td colspan="32">Data Payload (Variable)</td></tr>
<tr><td colspan="32">Trailer</td></tr>
</table>

Figure 1: Signal Data Packet Memory Layout

2.2.1 Header

- **Pkt Type** will be **0001** indicating it is a Signal Data packet with Stream Identifier.
- **C** will be set to **0** indicating there is no Class Identifier in this packet.
- **T** will be set to **1** indicating there is a trailer present. See the **Signal Data Packet Trailer Format** section.
- **N** will be set to **0** indicating expected successful parsing by a V49.0 parser.
- **S** will be set to **0** indicating that the signal data packet contains time domain IF data.
- **TSI** will be set to **01** to indicate the integer second field is in UTC time.
- **TSF** will be set to **10** to indicate the fractional second field is a real-time (picoseconds) timestamp.
- **Pkt Count** is a modulo-16 count of the signal data packets in the VRT packet stream. The count rolls over from **1111** to **0000**.
- **Packet Size** indicates the number of 32-bit words present in the signal data packet, including the header, payload, and trailer.

[1] *"VITA Radio Transport (VRT) Standard for Electromagnetic Spectrum: Signals and Applications" ANSI/VITA 49.2-2017, VITA Standard Organization, August 2017, http://www.vita.com/*

2.2.2 Stream Identifier

The stream identifier is a 32-bit number assigned to this VRT packet stream. All packets with this ID belong to the same stream. Stream IDs may be reused in a later program invocation, but are guaranteed to be unique between streams during a single program invocation.

2.2.3 Integer-seconds Timestamp

When the Integer-seconds timestamp conveys Coordinated Universal Time (UTC) it shall provide the reference point time of the first data sample in the packet in seconds since midnight January 6, 1980, Greenwich Mean Time.

2.2.4 Fractional-seconds Timestamp

The real-time timestamp extends the resolution of the Integer-seconds timestamp down to one picosecond. It accomplishes this by conveying the reference-point time of the first data sample in the Signal Data packet in picoseconds, relative to the time of the last Integer-seconds timestamp increment.

2.2.5 IF Data Payload

The IF Data Payload format for the Signal Data packet will be 16-bit signed integers representing complex Cartesian samples, or I/Q samples. Each I/Q data sample is a signed two's complement 16-bit value, ranging from -32768 to +32767.

<table>
<tr><th>Word</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td colspan="16">I1</td><td colspan="16">Q1</td></tr>
<tr><td>2</td><td colspan="16">I2</td><td colspan="16">Q2</td></tr>
<tr><td>3</td><td colspan="16">I3</td><td colspan="16">Q3</td></tr>
<tr><td>…</td><td colspan="32">…</td></tr>
<tr><td>N</td><td colspan="16">I(N)</td><td colspan="16">Q(N)</td></tr>
</table>

Figure 2: IF Data Payload Format

2.2.6 Trailer

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="12">Enables</td><td colspan="12">State/Event Indicators</td><td>E</td><td colspan="7">ACPC</td></tr>
</table>

Figure 3: Signal Data Packet Trailer Format

- Enable bits specify whether a specific functionality is present and indicator bits specify whether an event occurred related to that functionality, or specify the state of that functionality.
- If the enable bit is **0** then the indicator bit can be ignored.
- The **E** bit will be set to **1** to indicate the **Associated Context Packet Count (ACPC)** will be used.
- The **Associated Context Packet Count (ACPC)** will display a count of all transmitted Context packets that are associated with the Signal Data packet. This can also be thought of as the

number of Context packets that have been transmitted since the last change in a Context field (See Context Section).

Enable Bit Position	Indicator Bit Position	Indicator Name
31	19	Calibrated Time Indicator
30	18	Valid Data Indicator
29	17	Reference Lock Indicator
25	13	Over-Range Indicator
24	12	Sample Loss Indicator

Figure 4: Signal Data Packet Trailer Indicator Bits

2.2.6.1 Indicator Bits

- **Calibrated Time Indicator** specifies whether the timestamp in the IF data packet is calibrated to the external reference. A value of **one** specifies it is calibrated, a value of **zero** specifies it is free running and may not be accurate.
- The **Valid Data Indicator** specifies whether the data in the IF data packet is valid. When this bit is set to **zero,** some condition exists that may invalidate the data.
- The **Reference Lock Indicator** specifies whether the external reference input affecting the IF data is locked and stable. A value of **zero** indicates not external reference is present.
- The **Over-Range Indicator** specifies whether at least one sample in the IF data packet is invalid due to the signal exceeding the IF data range.
- The **Sample Loss Indicator** specifies whether the packet contains at least one sample discontinuity due to processing errors and/or buffer overflow.

2.3 Context Packet

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pkt Type				C	R	N	TSM	TSI		TSF		Pkt Count				Packet Size															
Stream Identifier																															
Integer Timestamp - Seconds																															
Fractional Timestamp - Picoseconds (Most significant word)																															
Fractional Timestamp - Picoseconds (Least significant word)																															
Context Section (variable length)																															

Figure 5: Context Packet Memory Layout

2.3.1 Header

- **Pkt Type** will be **0100** indicating it is a Context packet.
- **C** will be set to **0** indicating there is no Class Identifier in this packet.
- **R** is reserved and will be set to **0.**
- **N** will be set to **0** indicating expected successful parsing by a V49.0 parser.
- **TSM** will be set to **1** indicating that the timestamp is being used to convey general timing of events or Context changes.
- **TSI** will be set to **01** to indicate the integer second field is in UTC time.

- **TSF** will be set to **10** to indicate the fractional second field is a real-time (picoseconds) timestamp.
- **Pkt Count** is a modulo-16 count of the signal data packets in the VRT packet stream. The count rolls over from **1111** to **0000**.
- **Packet Size** indicates the number of 32-bit words present in the signal data packet, including the header, payload, and trailer.

2.3.2 Timestamping in the Context Packet

The timestamps in the Context packet will match the timestamps in a Signal Data packet in the paired Data Packet Stream and represents the reference point for the first sample in the data packet.

2.3.3 Context Section

The Context Section is a variable size field which contains a number of additional fields based on the contents of the Context Indicator Field. The Context Indicator Field is the first word in the Context Section and will always be present. This and all fields used by Signal Hound are discussed below.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">Context Indicator Field</td></tr>
<tr><td colspan="32">Context Fields (Present as indicated by the Context Indicators)</td></tr>
</table>

Figure 6: Context Section Format

- The first word of the Context Section is the Context Indicator Field, and every Context packet will contain it. The bits in the Context Indicator Field indicate which of the optional Context Fields are present in the Context packet, one for each field.
- The fields present in the Context Fields section are in the order as shown in Figure 7.

Figure 7 below details the position and definition of each bit in the context indicator field:

Bit Position	Context Indicator Field	Used	Number of Words in Context Field
31	Context Field Change Indicator	Yes	0
30	Reference Point Identifier	No	1
29	Bandwidth	Yes	2
28	IF Reference Frequency	No	2
27	RF Reference Frequency	Yes	2
26	RF Reference Frequency Offset	No	2
25	IF Band Offset	No	2
24	Reference Level	Yes	1
23	Gain	Yes	1
22	Over-Range Count	No	1
21	Sample Rate	Yes	2
20	Timestamp Adjustment	No	2
19	Timestamp Calibration Time	No	1
18	Temperature	Yes	1
17	Device Identifier	Yes	2
16	State and Event Indicators	No	1
15	IF Data Packet Payload Format	No	2
14	Formatted GPS Geolocation	Yes	11
13	Formatted INS Geolocation	No	11
12	ECEF (Earth-Centered, Earth-Fixed) Ephemeris	No	13
11	Relative Ephemeris	No	13
10	Ephemeris Reference Identifier	No	1
9	GPS ASCII	No	Variable
8	Context Association Lists	No	Variable
7..0	Reserved	No	N/A

Figure 7: Context Indicator Field Definitions

Only the fields that are marked **used** will be present in the Context packets and defined in this document.

2.3.3.1 Context Field Change Indicator

This bit is set to one when at least one Context Field contains a new value. If the Context Fields have not changed, the Context Field Change Indicator will be set to zero.

2.3.3.2 Bandwidth Field

The bandwidth field describes the amount of usable spectrum of the associated Signal Data Stream. The bandwidth is described by Signal Hound as the width in Hertz between the two cutoff frequencies of the IF bandpass filter. The cutoff frequencies are defined as the 3dB point, where the filter is attenuating the signal by 3dB. This value will always be smaller than the sample rate as the Signal Hound device always ensures the filter has sufficient amplitude roll off to prevent aliasing.

The value of the bandwidth field is expressed in Hertz and is in the 44.20 fixed point format as shown below.

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Bandwidth (63...32) , Hz																															
2	Bandwidth (31...20), Hz												Bandwidth (19...0), Hz (After Radix Point)																			

Figure 8: Bandwidth Context Field Format

2.3.3.3 RF Reference Frequency Field

The RF reference frequency field provides the original frequency the input was translated by to produce the IF data. This frequency is usually the frequency requested through device configuration. The RF reference frequency follows the format explained in the Frequency Format section.

2.3.3.4 Reference Level Field

The reference level field provides a power level that is used to calculate the magnitude of the received data. See Calculating I/Q Data Magnitude for an equation for this.

The reference level format is a 16-bit fixed point value of the format 9.7 as shown below.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																Ref Level - Integer									Ref Level - Fractional						

Figure 9: Reference Level Field Format

2.3.3.5 Gain/Attenuation Field

The gain/attenuation field describes the amount of attenuation the signal undergoes in the system, expressed in dB.

The attenuation values are formatted as shown below. Two 16-bit gain values in a 9.7 fixed point format are packed into one word.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Gain/Atten 2 (Unused)																Atten - Integer									Atten - Fractional						

Figure 10: Gain/Attenuation Context Field Format

2.3.3.6 Sample Rate Field

The sample rate field expresses the sample rate of the data samples in the paired Signal Data packet. The value of the field is expressed in Hertz and follows the format described in the Frequency Format section.

2.3.3.7 Temperature Field

The temperature field provides the internal operating temperature of the device. The temperature field is a 16-bit fixed point value expressed in Celsius units.

The fixed-point format is 10 integer bits and 6 fractional bits. The format is as shown below.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="10">Temp-Integer</td><td colspan="6">Temp-Frac</td></tr>
</table>

Figure 11: Temperature Context Field Format

2.3.3.8 Device Identifier Field

The device identifier field is used to identify the manufacturer and model of the device generating the Context Packet Stream.

The Manufacturer OUI field will be set to zero.

The Device code field will be set to 0x1000 for the SM200A.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="24">Manufacturer OUI (23..0)</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">Device Code (15..0)</td></tr>
</table>

Figure 12: Device Identifier Context Field Format

2.3.3.9 Formatted GPS Geolocation Field

The GPS geolocation context field format is displayed below:

<table>
<tr><td>Word</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="4">Reserved</td><td colspan="2">TSI</td><td colspan="2">TSF</td><td colspan="24">GPS Manufacturer OUI</td></tr>
<tr><td>2</td><td colspan="32">Integer-Second Timestamp of Position Fix (31..0)</td></tr>
<tr><td>3</td><td colspan="32">Fractional-Second Timestamp of Position Fix (63..32)</td></tr>
<tr><td>4</td><td colspan="32">Fractional-Second Timestamp of Position Fix (31..0)</td></tr>
<tr><td>5</td><td colspan="32">Latitude (31..0) degrees</td></tr>
<tr><td>6</td><td colspan="32">Longitude (31..0) degrees</td></tr>
<tr><td>7</td><td colspan="32">Altitude (31..0) meters</td></tr>
<tr><td>8</td><td colspan="32">Speed over Ground (31..0) meters/seconds</td></tr>
<tr><td>9</td><td colspan="32">Heading Angle (31..0) degrees</td></tr>
<tr><td>10</td><td colspan="32">Track Angle (31..0) degrees</td></tr>
<tr><td>11</td><td colspan="32">Magnetic Variation (31..0) degrees</td></tr>
</table>

Figure 13: Formatted GPS Geolocation Field Format

The GPS geolocation field is parsed from the GPRMC string from the internal GPS device.

Signal Hound uses only the following subfields: Integer-seconds timestamp, Fractional-second timestamps, Latitude, Longitude, and Altitude subfields.

The format used for the angle values represented in the GPS geolocation field are 32 bit fixed point values of the format 10.22 as shown below.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">Angle - Integer</td><td colspan="22">Angle - Fractional</td></tr>
</table>

Figure 14: Geolocation Angle Format

The **Latitude** subfield uses the angle format described and ranges from -90 (South) to +90 (North) degrees.

The **Longitude** subfield uses the angle format described and ranges from -180.0 (West) to +180.0 (East) degrees.

The **Altitude** format is a 32-bit fixed point value of the format 27.5 as shown below. The altitude format has a range of +-67108 km and a resolution of 3.1 centimeters.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="26">Altitude - Integer</td><td colspan="6">Alt - Frac</td></tr>
</table>

Figure 15: Altitude Format

The Latitude, Longitude, Altitude, Speed Over Ground, Heading, Track Angle, and Magnetic Variation fields will take the value of 0x7FFFFFFF when unspecified.

2.3.4 Frequency Format

A number of frequencies are of the format shown in Figure. The frequency is stored as a 64-bit fixed point number of the format 44.20, that is, an integer value represented in 44 bits, and a fractional value in 20 bits. This allows a frequency range of +/- 8.79 THz and a resolution of 0.95 μHz. The 64-bit value is split between two words, where the most significant word comes first.

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Frequency (63…32) , Hz																															
2	Frequency (31…20), Hz												Frequency (19…0), Hz (After Radix Point)																			

Figure 17: 64-bit Frequency Format

2.3.5 Calculating I/Q Data Magnitude

Each complex I/Q sample is represented as two consecutive values, and each value is processed identically when unpacking.

To unpack, each signed 16-bit integer value is converted to a floating point number between 0 and 1, and then scaled by the reference level. The equation is as follows:

$$x' = \frac{x}{2^{15}} \sqrt{10^{reflevel/10}}$$

In code, this is executed in a *for* loop, as follows:

$$\mathrm{dst} = (float)\mathrm{src}/pow(2,15) * sqrt(pow(10.0, \mathrm{reflevel}/10.0));$$

3 Usage

3.1 sm_api_vrt

To use the SM200A's VRT functionality, include *sm_api_vrt.h* in your project, in addition to *sm_api.h*.

VRT functions are defined in the SM200A API Manual.

3.2 sh_vrt

Structs, constants, useful functions, and other definitions used in working with VRT are provided in *sh_vrt.h* and *sh_vrt.cpp*. These are used by the API, the **VRTParser** class (see Parsing and GUI Examples), and can be included in user projects.

It is not necessary to include these files to use the VRT API functions.

4 Examples

A range of examples is provided in the VITA 49 directory of the SM API.

4.1 Basic

A simple example of VRT packet acquisition is demonstrated in *get_vrt_packets.cpp*.

4.2 Parsing

A full example of parsing a VRT Packet Stream consisting of Context and Signal Data packets is given using the **VRTParser** class in *vrt_test_parser_main.cpp*.

4.3 GUI

A small GUI parsing program built with QT and using the **VRTParser** class is provided.

5 Information & Support

5.1 Information

Release and meta information is available in the release notes file located in the top level of the SM series directory in the API download.

5.2 Support

The first troubleshooting step is always to power cycle the device.

For support contact support@signalhound.com.