SVS – Smart Vibration Sensor

User Manual

SVSK-00001, SVSK-00002
2026-09-17

1040, avenue Belvédère, suite 215, Québec, QC, G1S 3G3, Canada **T** 418-686-0993 **F** 418-686-2043 www.softdb.com

Soft dB

Table of Contents

1 Introduction 2

2 Installation 3

2.1 Connection 3

3 Programming 4

3.1 Serial Port Configuration 4

3.2 Protocol Description 4

3.3 Checksum 4

3.4 Initialization and Configuration 4

3.5 Hardware and Firmware Compatibility 5

3.6 Typical Applications 5

3.6.1 Real-Time Data 5
3.6.2 Average Data 6

3.7 Commands Description 6

3.7.1 Get Board Info 6
3.7.2 Get Sensor Info 7
3.7.3 Get Sensor Configuration 7
3.7.4 Set Sensor Configuration 8
3.7.5 Reset Buffer Status 9
3.7.6 Get Buffer Status 9
3.7.7 Get Data 10
3.7.8 Get Waveform 11
3.7.9 Get/Reset Average Data 12
3.7.10 Exec Sensor-Check 14
3.7.11 Get Sensor-Check 14
3.7.12 Get Temperature 15
3.7.13 Reset Device 15

1 Introduction

Congratulations for purchasing the SVS Smart Vibration Sensor. The SVS smart Vibration sensor is a Precise, Rugged and Versatile Embedded Vibration Meter.

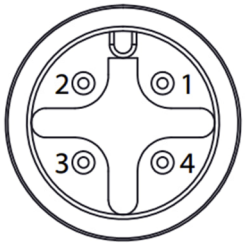
- 2.7kHz Sampling Rate;
- Frequency range 0.5 Hz – 1 kHz;
- DC measurement for accelerometer;
- 2 dynamic ranges (low and high);
- Complies to ISEE, DIN 45669-1 and ISO 8041;
- Automatic Sensor Check;
- Simple calibration with the gravity;
- Rugged and Compact Form-Factor;
- Digital communication avoiding electrical interferences;
- Up-To 100m cable length.

2 Installation

2.1 Connection

The SVS uses RS485 half-duplex serial communication. A shielded 4-pin M12 male connector is used for both digital communication and power-supply:

Pin	Description
1	RS-485 Pos.
2	RS-485 Neg.
3	+5V
4	Ground
Shell	Ground



The power-supply must provide at least 0.7A at 5V. The SVS peak current is 400mA and average current consumption is 100mA.

The maximum cable length between the host and the sensor is 100 meters.

Only one SVS sensor is possible on a single serial port: the daisy-chain architecture, where many sensors are on the same port, is not supported.

3 Programming

3.1 Serial Port Configuration

The SVS uses RS485 half-duplex serial communication with the following specifications:

- BaudRate: 3 MBPS
- WordLength: 8 bits
- StopBit: one bit
- Parity: No
- Control flow: No

3.2 Protocol Description

The SVS communication protocol is based on a set of commands and responses:

1) The host sends a command of 12 bytes to the SVS;
2) The SVS replies with the requested data and/or with an acknowledge.

The host must wait at least 5ms after receiving data from the SVS for the sensor to be ready to receive another command.

Whenever 16-bit integers, 32-bit integers or 32-bit floating point numbers are present in the command or payload, it must be represented with LSB-First topology (Least significant byte first).

3.3 Checksum

The protocol uses a simple checksum to validate the host commands and SVS data. Below is a code example to compute the checksum for a buffer of bytes:

```
unsigned char CheckSum = 0; // 8-bit unsigned
for (int i = 0; i < BytesBufferSize; i++)
{
      CheckSum = CheckSum ^ BytesBuffer[i];          // ^ operator is XOR
}
```

3.4 Initialization and Configuration

1) Configure the Host serial port (Serial Port Configuration);
2) Open the serial port socket;
3) The Host sends the Get Board Info command to validate that the connected sensor is an SVS.
4) The Host sends the Get Sensor Info command to determine the serial number of the SVS and validate the firmware version.

5) Set Sensor Configuration for your application. If the SVS is not configured, it will use the last configuration from its non-volatile memory.

3.5 Hardware and Firmware Compatibility

The Get Sensor Info command returns the SVS serial number and firmware version.

The SVS Model can be determined from the SVS serial number. Below is an example for serial number "VS0123102207":

- "VS": SVS
- "01": model ID "1"
- "231022": Manufacture batch ID
- "07": Specific ID in batch.

The current compatible firmware version is 10.

3.6 Typical Applications

The SVS may be used for real-time data acquisition or for average data acquisition. These two typical applications are detailed below.

It's also possible to combine these two modes of operation by performing real-time acquisition and average data acquisition in the same process.

3.6.1 Real-Time Data

For this application, the Host reads every data block and/or waveform block in real-time.

While the most common application is to read every data and waveform block, it's also possible to read every data block, but read only specific waveform blocks when a threshold is exceeded for example. Thanks to the SVS buffer, the host may read blocks in the buffer up to 9.3s in the past.

Reset Buffer

1) Reset Buffer Status to initialize the buffer (This step resets the block counter to zero);
2) Set a "Block ID" variable to zero.

Query Buffer Status

3) Periodically Get Buffer Status and compare the returned "Buffer Last Block ID" to the "Block ID" variable.

If the "Buffer Last Block ID" is greater than the "Block ID" variable, it means that blocks are ready to be read from the buffer.

The host may read any Block ID in the buffer. The available Block IDs are determined from the "Buffer Last Block ID" and the "Buffer Number of Blocks". Below is an example:

- Buffer Last Block ID = 842576
- Buffer Number of Blocks = 100
- Any Block ID from 842477 to 842576 can be read.

Read Blocks

4) Increment the "Block ID" variable;
5) Read the Block ID Data or Waveform using the following functions:
    - Get Data;
    - Get Waveform;

The host may read several blocks one after the other for as long as the "Buffer Last Block ID" is greater than the "Buffer Last Block ID".

3.6.2 Average Data

For this application, the Host reads average data on fixed intervals.

Reset Average

1) The Host resets the SVS running average by sending the Get/Reset Average Data command;

Wait for Interval

2) The Host waits for the average to complete using its own time reference.

Read Data and Reset Average

3) The Host reads the SVS running average by sending the Get/Reset Average Data command;

The Get/Reset Average Data command also resets the average. The Host may proceed directly to "Wait for Interval" for the next average.

3.7 Commands Description

3.7.1 Get Board Info

This command is the first command used to determine the presence of the sensor. The Vendor ID value must be 0x1612 and the Product ID must be 0x0108. The protocol sequence is:

The host sends the following 12-bytes command:

[0]	0x42
[1]	0x6F
[2]	0x61
[3]	0x72

[4]	0x64
[5]	0x49
[6]	0x6E
[7]	0x66
[8]	0x6F
[9]	0x2D
[10]	0x2D
[11]	Checksum

The SVS replies with 8 bytes:

[0-1]	Vendor ID (Unsigned 16-bit) (0x1612)
[2-3]	Product ID (Unsigned 16-bit) (0x0108)
[4-5]	Reserved
[6]	Checksum
[7]	Acknowledge (0x32)

3.7.2 Get Sensor Info

This command is used to read the serial number and firmware. This information is important to determine the Hardware and Firmware Compatibility.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x30
[4]	0x31
[5-10]	0x00
[11]	Checksum

The SVS replies with 17-bytes:

[0-11]	Serial Number (ASCII characters)
[12-13]	Firmware Version (Unsigned 16-bit)
[14]	Hardware Version (Unsigned 8-bit)
[15]	Checksum
[16]	Acknowledge (0x32)

3.7.3 Get Sensor Configuration

The current sensor configuration can be read back if desired.

The host sends the following 12-bytes command:

[0]	0x41

[1]	0x50
[2]	0x50
[3]	0x30
[4]	0x36
[5-10]	0x00
[11]	Checksum

The SVS replies with 61 bytes:

[0-5]	Configuration date (6x Unsigned 8-Bit (chars) "YYMMDD")
[6]	Input Range (Unsigned 8-bit) (0: Low, 1: High)
[7]	Fu frequency index for Acc. (Unsigned 8-bit) (0: 0.63Hz, 1: 0.8Hz, 2: 1.0Hz, 3: 1.25Hz, 4: 1.6Hz, 5: 2.0Hz, 6: 2.5Hz, 7: 3.15Hz, 8: 4.0Hz, 9: 5.0Hz, 10: 6.3Hz)
[8]	Fo frequency index for Acc. (Unsigned 8-bit) (0: 80Hz, 1: 100Hz, 2: 125Hz, 3: 160Hz, 4: 200Hz, 5: 250Hz, 6: 315Hz, 7: 400Hz, 8: 500Hz, 9: 630Hz, 10: 800Hz)
[9]	Fu frequency index for Vel. (Unsigned 8-bit) (0: 0.63Hz, 1: 0.8Hz, 2: 1.0Hz, 3: 1.25Hz, 4: 1.6Hz, 5: 2.0Hz, 6: 2.5Hz, 7: 3.15Hz, 8: 4.0Hz, 9: 5.0Hz, 10: 6.3Hz)
[10]	Fo frequency index for Vel. (Unsigned 8-bit) (0: 80Hz, 1: 100Hz, 2: 125Hz, 3: 160Hz, 4: 200Hz, 5: 250Hz, 6: 315Hz, 7: 400Hz, 8: 500Hz, 9: 630Hz, 10: 800Hz)
[11-58]	Reserved
[59]	Checksum
[60]	Acknowledge (0x32)

3.7.4 Set Sensor Configuration

This command allows to configured the input range and the low and high frequency of the sensor bandwidth response for both acceleration and velocity to match the desired measurement standard.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x30
[4]	0x37
[5-10]	0x00
[11]	Checksum

The SVS responds with 1 byte:

[0]	Acknowledge (0x32)

The host waits 5ms.

The host sends the following 60-bytes configuration:

[0-5]	Configuration date (6x Unsigned 8-Bit (chars) "YYMMDD")

[6]	Input Range (Unsigned 8-bit) (0: Low, 1: High)
[7]	Fu frequency index for Acc. (Unsigned 8-bit) (0: 0.63Hz, 1: 0.8Hz, 2: 1.0Hz, 3: 1.25Hz, 4: 1.6Hz, 5: 2.0Hz, 6: 2.5Hz, 7: 3.15Hz, 8: 4.0Hz, 9: 5.0Hz, 10: 6.3Hz)
[8]	Fo frequency index for Acc. (Unsigned 8-bit) (0: 80Hz, 1: 100Hz, 2: 125Hz, 3: 160Hz, 4: 200Hz, 5: 250Hz, 6: 315Hz, 7: 400Hz, 8: 500Hz, 9: 630Hz, 10: 800Hz)
[9]	Fu frequency index for Vel. (Unsigned 8-bit) (0: 0.63Hz, 1: 0.8Hz, 2: 1.0Hz, 3: 1.25Hz, 4: 1.6Hz, 5: 2.0Hz, 6: 2.5Hz, 7: 3.15Hz, 8: 4.0Hz, 9: 5.0Hz, 10: 6.3Hz)
[10]	Fo frequency index for Vel. (Unsigned 8-bit) (0: 80Hz, 1: 100Hz, 2: 125Hz, 3: 160Hz, 4: 200Hz, 5: 250Hz, 6: 315Hz, 7: 400Hz, 8: 500Hz, 9: 630Hz, 10: 800Hz)
[11-58]	Reserved (Use bytes [11-58] from the Get Sensor Configuration function)
[59]	Checksum

The SVS responds with 1 byte:

[0]	Acknowledge (0x32)

3.7.5 Reset Buffer Status

This command is used at the beginning of the acquisition session or when desired to empty the acquisition buffer: 100 blocks of 93ms for both the data and the time signal are erased.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x30
[4]	0x33
[5-10]	0x00
[11]	Checksum

The SVS replies with 1 byte:

[0]	Acknowledge (0x32)

3.7.6 Get Buffer Status

This command is used to determine the number of blocks in the buffer for both the Data and Waveform. Each block has a duration of 93ms and there are 100 blocks in the buffer for a total of 9.3s.

The buffer status indicates the quantity of blocks available in the buffer as well as the ID of the last block in the buffer.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50

[2]	0x50
[3]	0x30
[4]	0x32
[5-10]	0x00
[11]	Checksum

The SVS replies with 18-bytes:

[0-3]	Buffer Number of Blocks (Unsigned 32-bit)
[4-11]	Reserved
[12-15]	Buffer Last Block ID (Unsigned 32-bit)
[16]	Checksum
[17]	Acknowledge (0x32)

3.7.7 Get Data

This command allows to get a specific instant data block of 93ms in the sensor buffer. This command is useful to monitor the peak value to trig a time signal recording.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x30
[4]	0x34
[5-8]	Block ID (Unsigned 32-bit)
[9-10]	0x00
[11]	Checksum

The SVS replies with 138 bytes:

[0-3]	Block ID (Unsigned 32-bit)
[4-7]	Acc. DC X, m/s^2 (Float 32-bit)
[8-11]	Acc. DC Y, m/s^2 (Float 32-bit)
[12-16]	Acc. DC Z, m/s^2 (Float 32-bit)
[16-19]	Acc. DC Vector Sum, m/s^2 (Float 32-bit)
[20-23]	Acc. RMS X, m/s^2 (Float 32-bit)
[24-27]	Acc. RMS Y, m/s^2 (Float 32-bit)
[28-31]	Acc. RMS Z, m/s^2 (Float 32-bit)
[32-35]	Acc. RMS Vector Sum, m/s^2 (Float 32-bit)
[36-39]	Acc. Peak X, m/s^2 (Float 32-bit)
[40-43]	Acc. Peak Y, m/s^2 (Float 32-bit)
[44-47]	Acc. Peak Z, m/s^2 (Float 32-bit)
[48-51]	Acc. Peak, Vector Sum, m/s (Float 32-bit)
[52-55]	Acc. RMS ISO X, m/s^2 (Float 32-bit)
[56-59]	Acc. RMS ISO Y, m/s^2 (Float 32-bit)

[60-63]	Acc. RMS ISO Z, m/s^2 (Float 32-bit)
[64-67]	Acc. RMS ISO Vector Sum, m/s^2 (Float 32-bit)
[68-71]	Acc. Peak ISO X, m/s^2 (Float 32-bit)
[72-75]	Acc. Peak ISO Y, m/s^2 (Float 32-bit)
[76-79]	Acc. Peak ISO Z, m/s^2 (Float 32-bit)
[80-83]	Acc. Peak ISO Vector Sum, m/s (Float 32-bit)
[84-87]	Vel. RMS X, m/s (Float 32-bit)
[88-91]	Vel. RMS Y, m/s (Float 32-bit)
[92-95]	Vel. RMS Z, m/s (Float 32-bit)
[96-99]	Vel. RMS Vector Sum, m/s (Float 32-bit)
[100-103]	Vel. Peak X, m/s (Float 32-bit)
[104-107]	Vel. Peak Y, m/s (Float 32-bit)
[108-111]	Vel. Peak Z, m/s (Float 32-bit)
[112-115]	Vel. Peak Vector Sum, m/s (Float 32-bit)
[116-119]	Vel. KBFmax, X, m/s (Float 32-bit)
[120-123]	Vel. KBFmax, Y, m/s (Float 32-bit)
[124-127]	Vel. KBFmax, Z, m/s (Float 32-bit)
[128-131]	Vel. KBFmax, Vector Sum, m/s (Float 32-bit)
[132]	Status, X (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload, Bit2: Under-Range Acc., Bit3: Under-Range Vel.)
[133]	Status, Y (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload, Bit2: Under-Range Acc., Bit3: Under-Range Vel.)
[134]	Status, Z (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload, Bit2: Under-Range Acc., Bit3: Under-Range Vel.)
[135]	Status, Vector Sum (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload, Bit2: Under-Range Acc., Bit3: Under-Range Vel.)
[136]	Checksum
[137]	Acknowledge (0x32)

3.7.8 Get Waveform

This command allows to get a specific instant time signal block of 93ms in the sensor buffer.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x30
[4]	0x35
[5-8]	Block ID (Unsigned 32-bit)
[9-10]	0x00
[11]	Checksum

The SVS replies with 3106 bytes:

[0-3]	Block ID (Unsigned 32-bit)

[4]	Status, X (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload)
[5]	Status, Y (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload)
[6]	Status, Z (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload)
[7]	Status, Vector Sum (Bit0: Reserved, Bit1: Overload)
[8-11]	Acc. Waveform Scaling factor, X (Float 32-bit)
[12-15]	Acc. Waveform Scaling factor, Y (Float 32-bit)
[16-19]	Acc. Waveform Scaling factor, Z (Float 32-bit)
[20-531]	Acc. Waveform, X (256x Signed 16-bit)
[532-1043]	Acc. Waveform, Y (256x Signed 16-bit)
[1044-1555]	Acc. Waveform, Z (256x Signed 16-bit)
[1556-1559]	Vel. Waveform Scaling factor, X (Float 32-bit)
[1560-1563]	Vel. Waveform Scaling factor, Y (Float 32-bit)
[1564-1567]	Vel. Waveform Scaling factor, Z (Float 32-bit)
[1568-2079]	Vel. Waveform, X (256x Signed 16-bit)
[2080-2591]	Vel. Waveform, Y (256x Signed 16-bit)
[2592-3103]	Vel. Waveform, Z (256x Signed 16-bit)
[3104]	Checksum
[3105]	Acknowledge (0x32)

Multiply the "Scaling Factor" (Float 32-bit) with the digital waveform values (Signed 16-bit, -32768 to 32767) to represent the waveform in m/s or m/s^2.

3.7.9 Get/Reset Average Data

The SVS automatically computes an average. The same function is used to read the accumulated data and reset the average.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x31
[4]	0x38
[5-10]	0x00
[11]	Checksum

The SVS replies with 222 bytes and resets the average:

[0-3]	Vel. Peak, X, m/s (Float 32-bit)
[4-7]	Vel. Peak, Y, m/s (Float 32-bit)
[8-11]	Vel. Peak, Z, m/s (Float 32-bit)
[12-15]	Vel. Peak Zero-Crossing Pseudo-Frequency, X, Hz (Float 32-bit)
[16-19]	Vel. Peak Zero-Crossing Pseudo-Frequency, Y, Hz (Float 32-bit)
[20-23]	Vel. Peak Zero-Crossing Pseudo-Frequency, Z, Hz (Float 32-bit)
[24-27]	Vel. Time of Peak, X, s (Float 32-bit)
[28-31]	Vel. Time of Peak, Y, s (Float 32-bit)

[32-35] Vel. Time of Peak, Z, s (Float 32-bit)
[36-39] Average Duration, s (Float 32-bit)
[40-43] Acc. DC X, m/s^2 (Float 32-bit)
[44-47] Acc. DC Y, m/s^2 (Float 32-bit)
[48-51] Acc. DC Z, m/s^2 (Float 32-bit)
[52-55] Acc. DC Vector Sum, m/s^2 (Float 32-bit)
[56-59] Acc. RMS X, m/s^2 (Float 32-bit)
[60-63] Acc. RMS Y, m/s^2 (Float 32-bit)
[64-67] Acc. RMS Z, m/s^2 (Float 32-bit)
[68-71] Acc. RMS Vector Sum, m/s^2 (Float 32-bit)
[72-75] Acc. Peak X, m/s^2 (Float 32-bit)
[76-79] Acc. Peak Y, m/s^2 (Float 32-bit)
[80-83] Acc. Peak Z, m/s^2 (Float 32-bit)
[84-87] Acc. Peak, Vector Sum, m/s^2 (Float 32-bit)
[88-91] Acc. RMS ISO X, m/s^2 (Float 32-bit)
[92-95] Acc. RMS ISO Y, m/s^2 (Float 32-bit)
[96-99] Acc. RMS ISO Z, m/s^2 (Float 32-bit)
[100-103] Acc. RMS ISO Vector Sum, m/s^2 (Float 32-bit)
[104-109] Acc. Peak ISO X, m/s^2 (Float 32-bit)
[108-111] Acc. Peak ISO Y, m/s^2 (Float 32-bit)
[112-115] Acc. Peak ISO Z, m/s^2 (Float 32-bit)
[116-119] Acc. Peak ISO Vector Sum, m/s^2 (Float 32-bit)
[120-123] Vel. RMS X, m/s (Float 32-bit)
[124-127] Vel. RMS Y, m/s (Float 32-bit)
[128-131] Vel. RMS Z, m/s (Float 32-bit)
[132-135] Vel. RMS Vector Sum, m/s (Float 32-bit)
[136-139] Vel. Peak X, m/s (Float 32-bit)
[140-143] Vel. Peak Y, m/s (Float 32-bit)
[144-147] Vel. Peak Z, m/s (Float 32-bit)
[148-151] Vel. Peak Vector Sum, m/s (Float 32-bit)
[152-155] Vel. KBFmax, X, m/s (Float 32-bit)
[156-159] Vel. KBFmax, Y, m/s (Float 32-bit)
[160-163] Vel. KBFmax, Z, m/s (Float 32-bit)
[164-167] Vel. KBFmax, Vector Sum, m/s (Float 32-bit)
[168] Status, X (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload, Bit2: Under-Range Acc., Bit3: Under-Range Vel.)
[169] Status, Y (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload, Bit2: Under-Range Acc., Bit3: Under-Range Vel.)
[170] Status, Z (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload, Bit2: Under-Range Acc., Bit3: Under-Range Vel.)
[171] Status, Vector Sum (Unsigned 8-bit) (Bit0: Reserved, Bit1: Overload, Bit2: Under-Range Acc., Bit3: Under-Range Vel.)
[172-175] Acc. RMSmax, X, m/s^2 (Float 32-bit)
[176-179] Acc. RMSmax, Y, m/s^2 (Float 32-bit)
[180-183] Acc. RMSmax, Z, m/s^2 (Float 32-bit)
[184-187] Acc. RMSmax, Vector Sum, m/s^2 (Float 32-bit)

[188-191]	Acc. RMSmax ISO, X, m/s^2 (Float 32-bit)
[192-195]	Acc. RMSmax ISO, Y, m/s^2 (Float 32-bit)
[196-199]	Acc. RMSmax ISO, Z, m/s^2 (Float 32-bit)
[200-203]	Acc. RMSmax ISO, Vector Sum, m/s^2 (Float 32-bit)
[204-207]	Vel. RMSmax, X, m/s (Float 32-bit)
[208-211]	Vel. RMSmax, Y, m/s (Float 32-bit)
[212-215]	Vel. RMSmax, Z, m/s (Float 32-bit)
[216-219]	Vel. RMSmax, Vector Sum, m/s (Float 32-bit)
[220]	Checksum
[221]	Acknowledge (0x32)

3.7.10 Exec Sensor-Check

This command executes a sensor-check on the SVS.

The sensor-check performs a test and the result is compared to a factory reference for compliance assessment.

The host must provide the current Unix time reference in the command for the SVS to add the time reference to the sensor-check.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x31
[4]	0x30
[5-10]	Unix Time Reference (Unsigned 64-bit, least significant 6 bytes)
[11]	Checksum

The SVS replies with an acknowledge byte after about 3s when the sensor-check is completed:

[0]	Acknowledge (0x32)

3.7.11 Get Sensor-Check

This command allows to read the sensor-check result from the SVS non-volatile memory.

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x31
[4]	0x31
[5-9]	0x00

[10]	0x00:
[11	Checksum

The SVS replies with 18 bytes:

[0]	Valid test (Unsigned 8-bit) 0: No 1: Yes
[1-4]	Acc. Measure X, m/s^2 (Float 32-bit)
[5-8]	Acc. Measure Y, m/s^2 (Float 32-bit)
[9-12]	Acc. Measure Z, m/s^2 (Float 32-bit)
[13]	Conformity X (Unsigned 8-bit) (0: Fail 1: Pass)
[14]	Conformity Y (Unsigned 8-bit) (0: Fail 1: Pass)
[15]	Conformity Z (Unsigned 8-bit) (0: Fail 1: Pass)
[16]	Checksum
[17]	Acknowledge (0x32)

3.7.12 Get Temperature

This command allows to obtain the current temperature in Celsius for both the processor (MCU) and the accelerometer of the SVS. The protocol sequence is:

The host sends the following 12-bytes command:

[0]	0x41
[1]	0x50
[2]	0x50
[3]	0x30
[4]	0x39
[5-11]	0x00
[11]	Checksum

The SVS replies with 10 bytes:

[0-3]	Microcontroller Temperature, °C (Float 32-bit)
[4-7]	Accelerometer Temperature, °C (Float 32-bit)
[8]	Checksum
[9]	Acknowledge (0x32)

3.7.13 Reset Device

This command can be used to make a software reset of the SVS.

The host sends the following 12-bytes command:

[0]	0x42
[1]	0x6F
[2]	0x61
[3]	0x72

[4]	0x64
[5]	0x52
[6]	0x65
[7]	0x73
[8]	0x65
[9]	0x74
[10]	0x2D
[11]	Checksum

The SVS does not reply to this command and will be ready for a new command after about 2s.