MEASURAND

SAAXYZ User Manual

Exported on 02/23/2021

Table of Contents

1 Introduction

The SAAXYZ is a simple microprocessor interface to Measurand's ShapeAccelArray (SAA) devices. It allows you to get Cartesian XYZ position data directly out of the SAA.

The two figures below illustrate how the SAAXYZ can be used with either a single SAA or up to five SAAs at once.

This manual describes the ASCII text commands that are used for controlling, getting status information, and getting data from Measurand's SAAXYZ device.

This manual describes the ASCII text commands that are used for controlling, getting status information, and getting data from Measurand's SAAXYZ device.

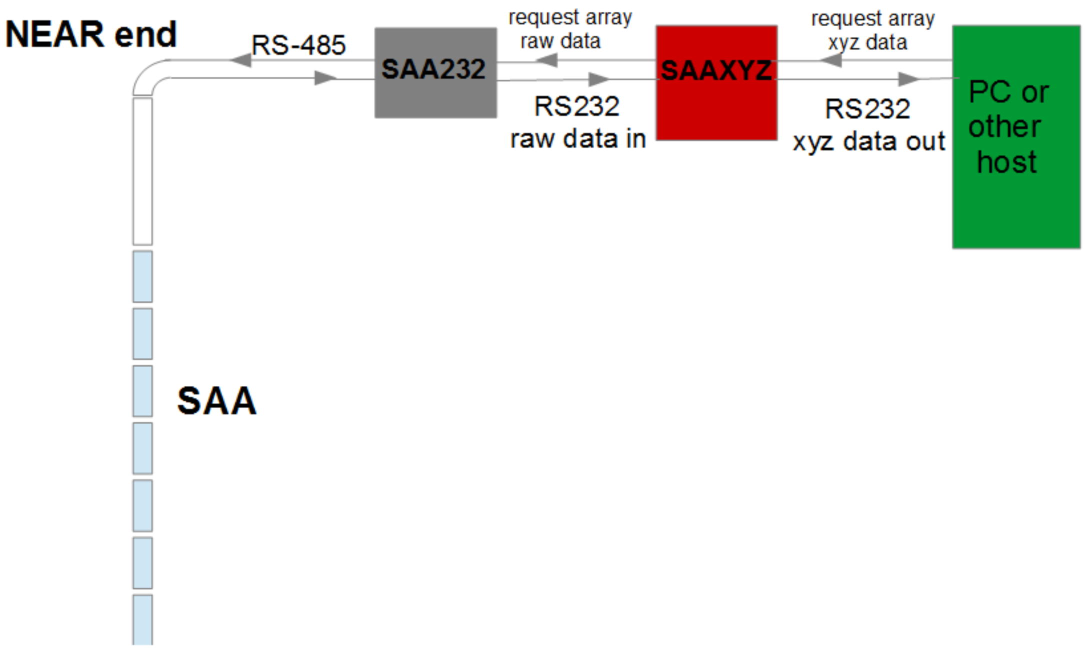


Figure 1: Diagram illustrating how the SAAXYZ is used with a single SAA. An SAA232 is required for controlling power to the SAA, surge protection, and RS-232 to RS-485 conversion

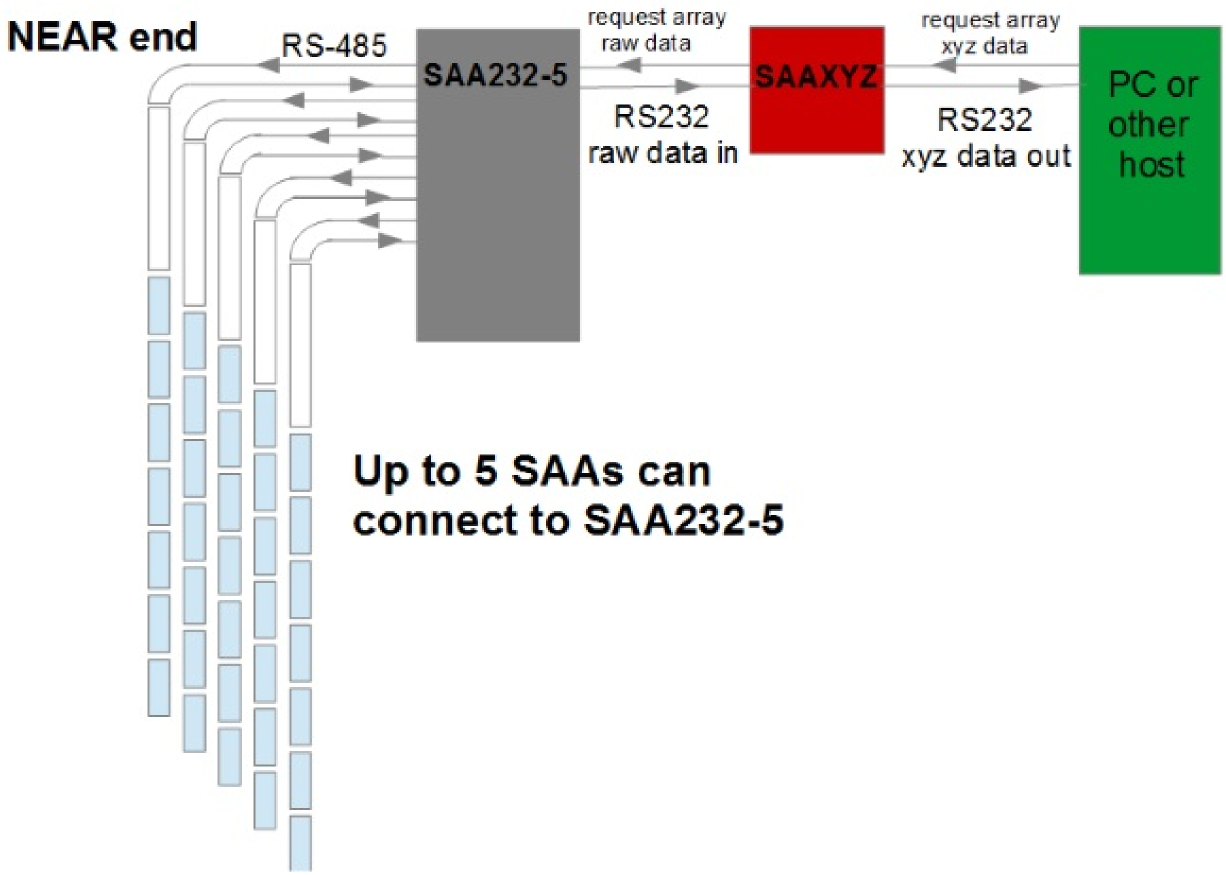


Figure 2: Diagram illustrating how the SAAXYZ is used with multiple (up to five) SAAs. An SAA232-5 is required for switching power to the SAAs, surge protection, and RS-232 to RS-485 conversion

2 Hardware Setup

1. Unplug all USB Serials from PC.
2. Connect SAAXYZ USB cable to SAAXYZ and to the +12 V power adaptor.

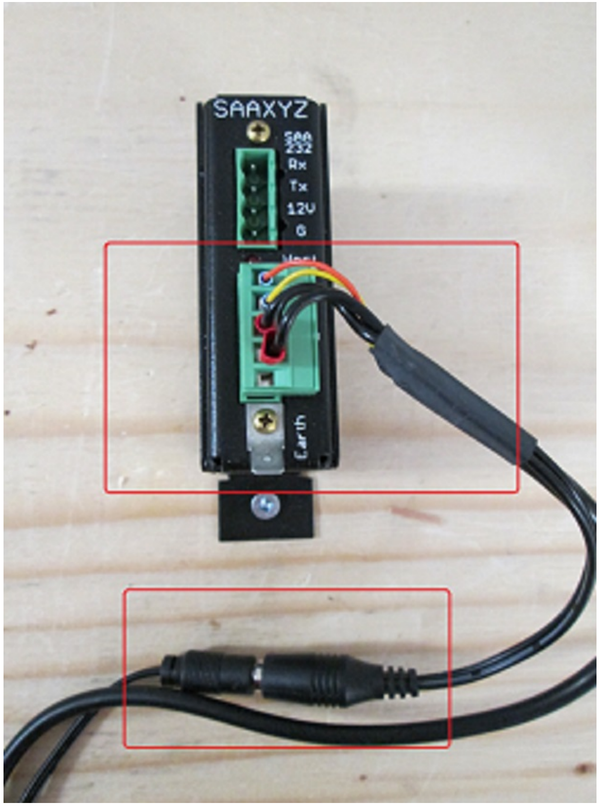


Figure 3: SAAXYZ with power supply

3. Connect the 4-pin Phoenix to Phoenix connector from SAAXYZ to SAA232(-5).
4. Connect SAA to SAA232. Use an SAA232-5 for more than one SAA.
5. Attach SAAXYZ USB Cable to PC.

3 SAAXYZ Setup

In order to install new calibration files onto the SAAXYZ or upgrade the firmware, the “SAAXYZ Setup” software must be installed.

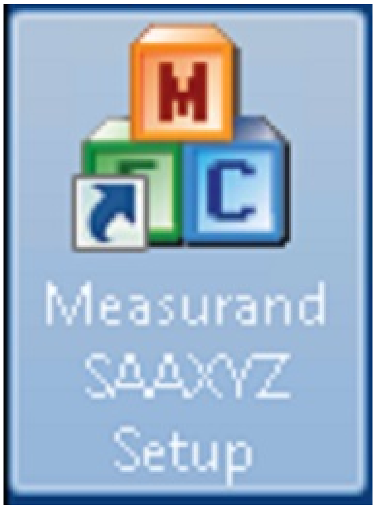
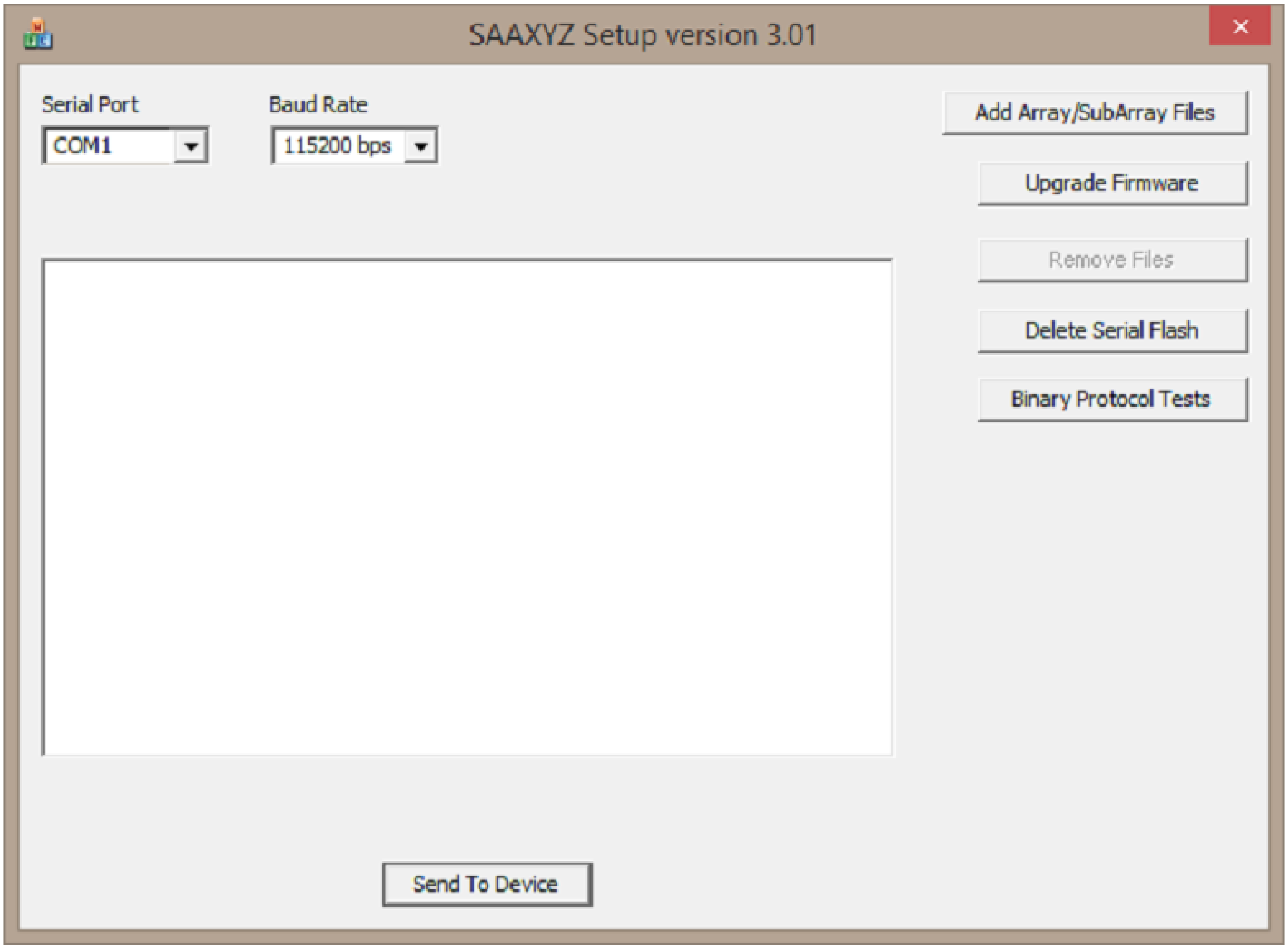


Figure 4: SAAXYZ Setup Software Icon

> ⓘ **NOTE**: The Terminal Software (e.g. TeraTerm) should not be run at the same time that the SAAXYZ Setup Program is running since they use the same serial link to the SAAXYZ device.



SAAXYZ Setup version 3.01

Serial Port: COM1

Baud Rate: 115200 bps

Figure 5: SAAXYZ Setup software. Used to upload calibration files or new firmware to the SAAXYZ

3.1 Add Array / Subarray Files

The Add Array / Subarray Files button will add Array and Subarray files to the SAAXYZ.

Before you begin, you must have your calibration files and FTDI communications drivers. To do this use SAASuite[1] to download SAARecorder, ensure that you follow through the whole process allowing the FTDI communications drivers to download. Also, download calibration files using SAASuite.

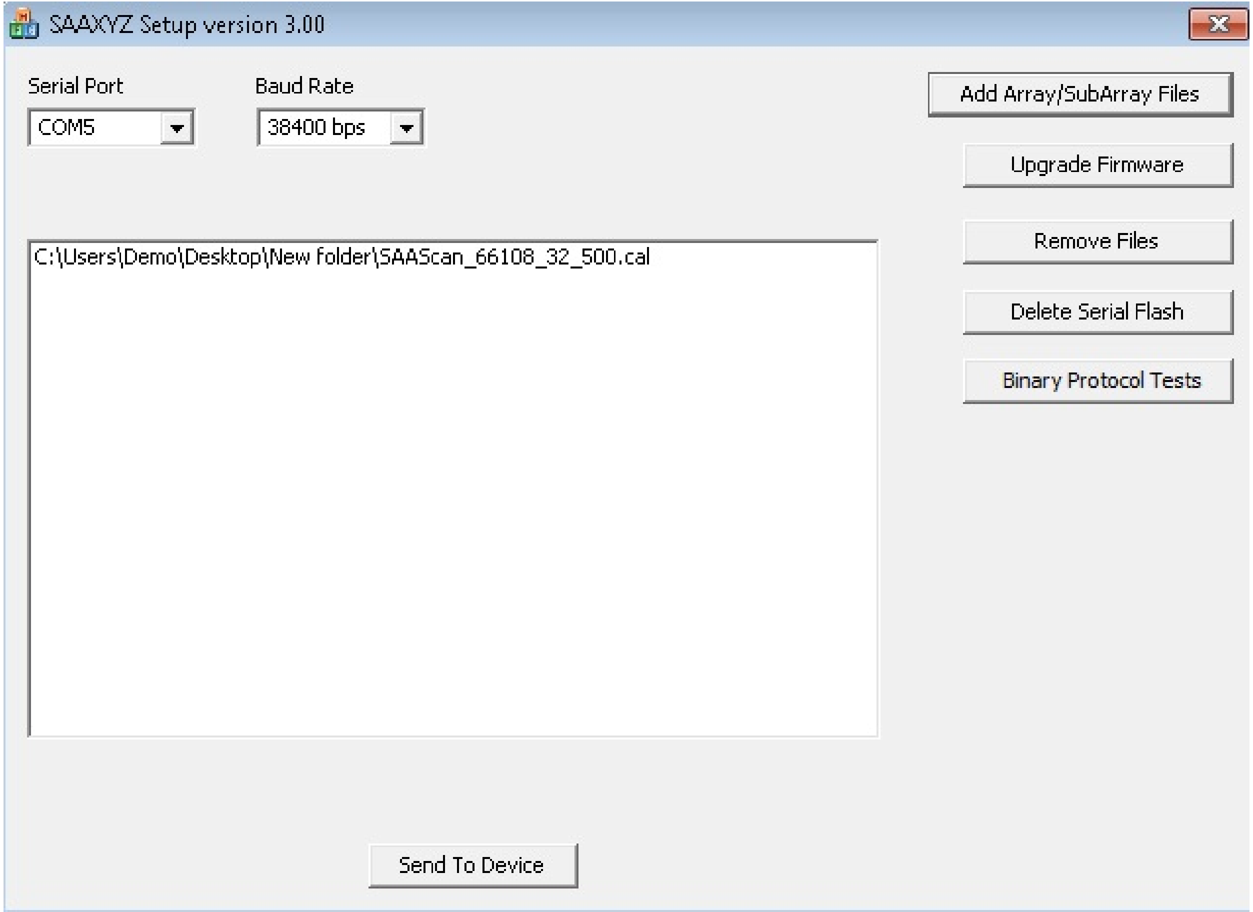
If the SAAXYZ is being used with a new SAA, click the 'Add Array/SubArray Files' button to select the "Array" calibration file for that SAA. SAA's with serial numbers higher than 66000 (i.e. Model 3 and up) only require an array calibration file (no Subarray files). Then click the 'Send To Device' button to send the file to the SAAXYZ.

[1] https://support.measurand.com/solution/articles/19000027451-how-to-download-and-install-saasuite

ⓘ **NOTE**: Ensure the device is setup with appropriate power, COM Port, and Baud Rate.

ⓘ **NOTE**: It is recommended that Array and Subarray files be copied into their own folder.

1. Click ‘Add Array/SubArray Files’ button. A navigation window will display.
2. Navigate to location of the file(s) and select all the ".cal" files related to the SAA:
   - Array files are always required
   - Subarrays are only required for SAAs with Serial Numbers less than 66000
   - Default location should be 'C:\measurand inc\saarecorder\calibrations'
3. Selected files will display in the Main window


SAAXYZ Setup version 3.00

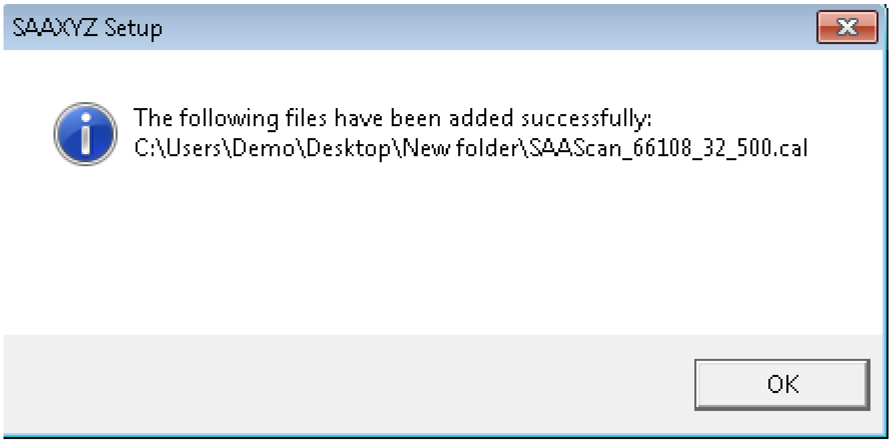
Serial Port: COM5

Baud Rate: 38400 bps

C:\Users\Demo\Desktop\New folder\SAAScan_66108_32_500.cal

Figure 6: Main window with selected '.cal' files

4. When all ".cal" files have been selected, click ‘Send to Device'. A Success window will pop up when complete.



SAAXYZ Setup

The following files have been added successfully:
C:\Users\Demo\Desktop\New folder\SAAScan_66108_32_500.cal

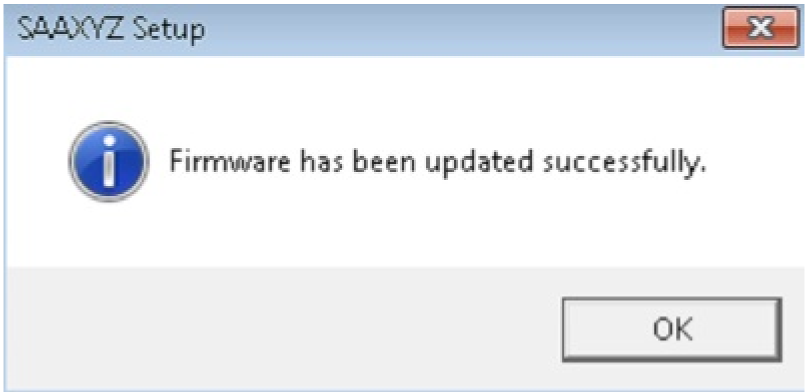
Figure 7: Success window for adding files

5. Unplug power from device, wait 10 seconds, and then reconnect the power.

3.2 Upgrade Firmware

The Upgrade Firmware button will allow the SAAXYZ to have the most current software associated with it.

1. Click ‘Upgrade Firmware’ button.
2. Navigate to firmware location.
3. Select ".bin" file and click ‘Open’.
4. Wait for Success window to pop up.



SAAXYZ Setup

Firmware has been updated successfully.

Figure 8: Success window for updating firmware

3.3 Remove Files

Clicking on the Remove Files button will remove any files located on the main window.

ⓘ **NOTE**: This does not remove files from the SAAXYZ.

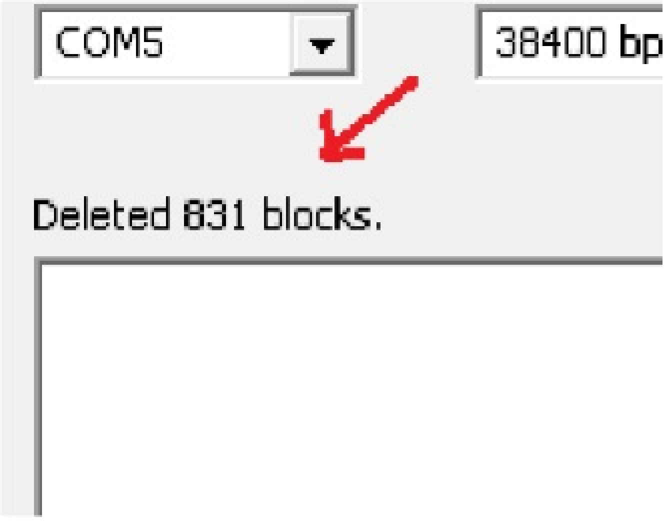
3.4 Delete Serial Flash

The Delete Serial Flash button will remove any previously entered Array / Subarray files from the SAAXYZ. Measurand recommends the use of this feature for new Projects.

If old calibration files are present on the SAAXYZ, they should be removed using this button, prior to installing the new calibration files. It takes approximately 2 minutes to delete the entire serial flash of the SAAXYZ.

ⓘ **NOTE**: This will remove the files from the SAAXYZ, not the Main window.

1. Click ‘Delete Serial Flash’.
2. Counter will show progress, see figure below.



COM5

38400 bp

Deleted 831 blocks.

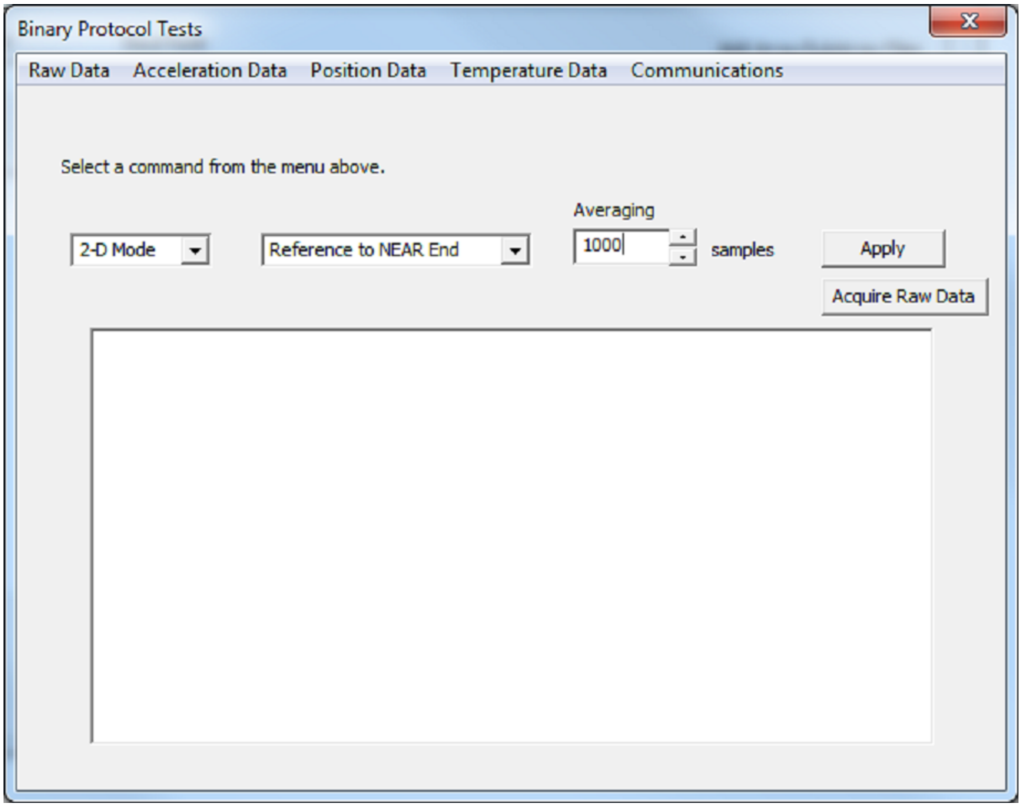
Figure 9: File counter on the Main window

3. Unplug power from device, wait 10 seconds, and then reconnect the power to ensure removal of file(s)

3.5 Binary Protocol Tests

The Binary Protocol Tests button demonstrates that the SAAXYZ has been setup properly by collecting data. These tests are based upon Binary Protocols, verifying all the commands work.

NOTE: None of the menu commands in the 'Binary Protocol Tests' window will work if 'Acquire Raw Data' has not been clicked first. Mode, Reference, and Averaging must be set to acquire raw data. Also, the Mode must be set to 3-D or 2-D (Horizontal). 2-D convergence mode is not supported.



Select a command from the menu above.

2-D Mode

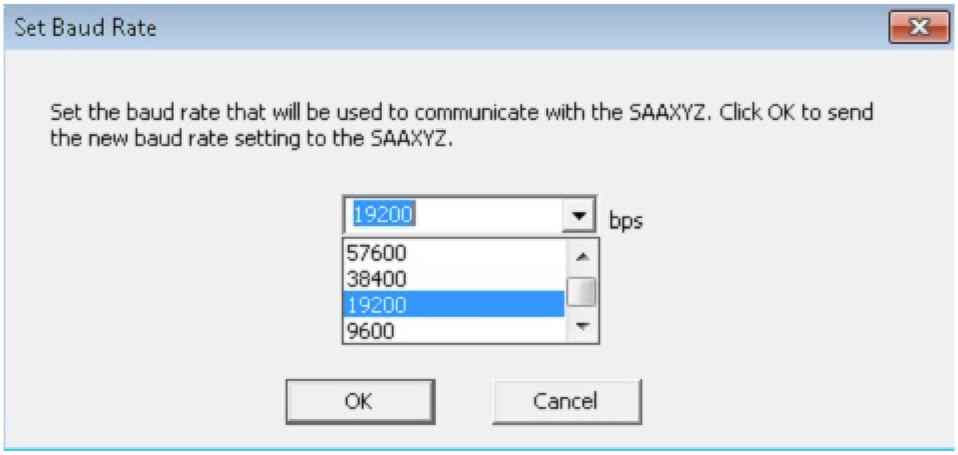
Reference to NEAR End

Averaging 1000 samples

Figure 10: Binary Protocol Tests window

3.5.1 Communications | Set Baud Rate

The Set Baud Rate option in the Communications menu is used to set the baud rate of the SAAXYZ.



Set Baud Rate

Set the baud rate that will be used to communicate with the SAAXYZ. Click OK to send the new baud rate setting to the SAAXYZ.

19200 bps

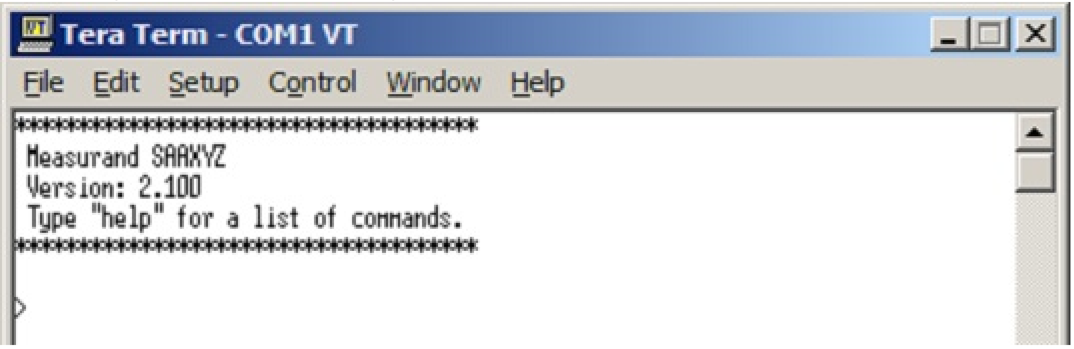
Figure 11: Baud Rate selection window

4 Steps for Using SAAXYZ

For testing, a terminal program on a PC ("TeraTerm[2]" is a nice reliable terminal program) could be used to test out the connection to the SAAXYZ.

The PC serial port should be configured for 38400 bps, 8 data bits, no parity, one stop bit, and no hardware handshaking.

1. Connect short 4-pin Phoenix to Phoenix connector from SAAXYZ to SAA232 if a single SAA is being used, or to the SAA232-5 if more than one SAA is being used.
2. Start terminal software on PC.
3. Plug +12 V power into SAAXYZ.
4. Something similar to the following in the terminal software should be observed.



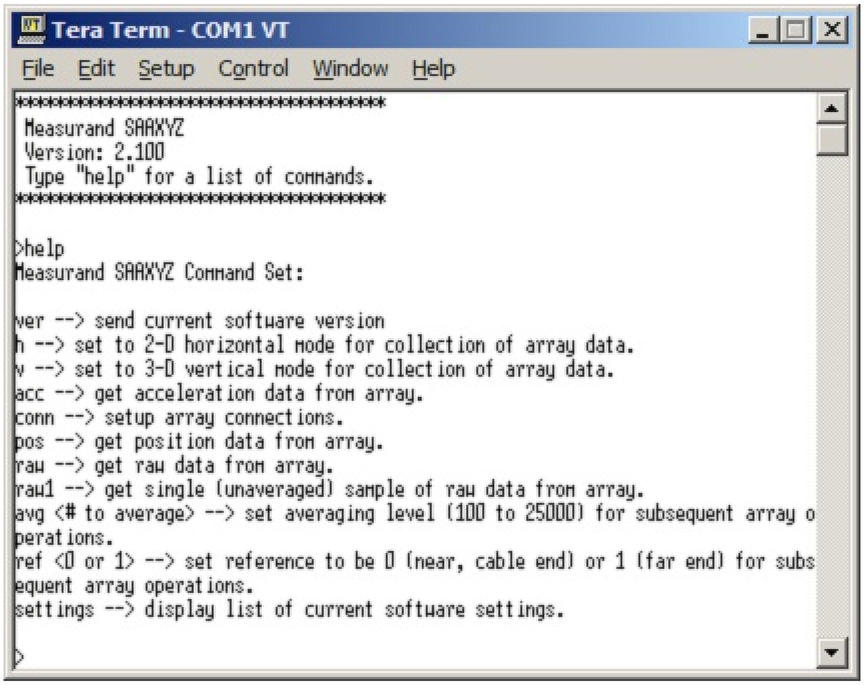
```
*************************************
 Measurand SAAXYZ
 Version: 2.100
 Type "help" for a list of commands.
*************************************


>
```

Figure 12: Terminal output from SAAXYZ after power up

5. Type 'help' for a list of commands.

[2] http://www.tucows.com/preview/195282



```
****************************************
 Measurand SAAXYZ
 Version: 2.100
 Type "help" for a list of commands.
****************************************

>help
Measurand SAAXYZ Command Set:

ver --> send current software version
h --> set to 2-D horizontal mode for collection of array data.
v --> set to 3-D vertical mode for collection of array data.
acc --> get acceleration data from array.
conn --> setup array connections.
pos --> get position data from array.
raw --> get raw data from array.
raw1 --> get single (unaveraged) sample of raw data from array.
avg <# to average> --> set averaging level (100 to 25000) for subsequent array o
perations.
ref <0 or 1> --> set reference to be 0 (near, cable end) or 1 (far end) for subs
equent array operations.
settings --> display list of current software settings.

>
```

Figure 13: List of commands available in SAAXYZ version 2.100

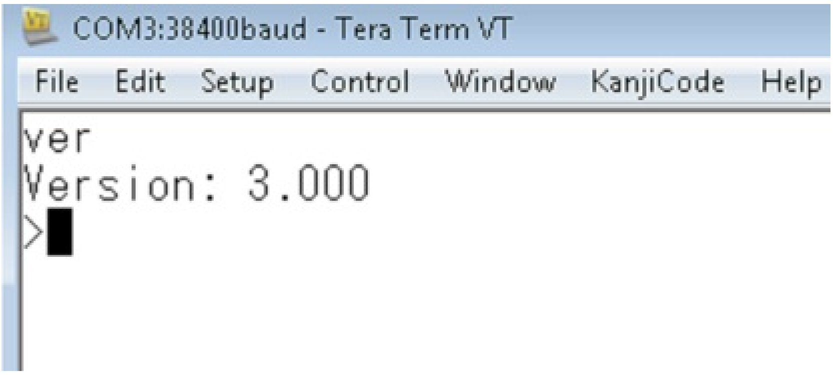
5 SAAXYZ Commands

The following lists the output that can be expected for each of the SAAXYZ commands.

NOTE: Each command string must be followed by a carriage return and/or linefeed character.

5.1 ver

The ver command sends the current SAAXYZ software version. The output will be an ASCII string of the form shown in the figure below.

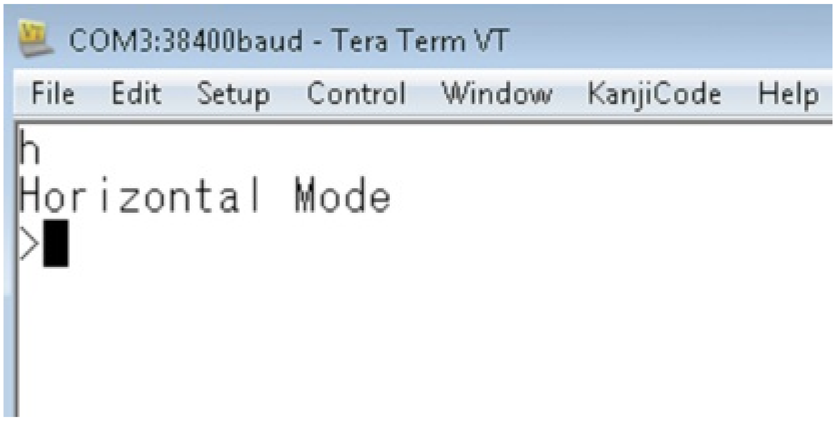

COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
ver
Version: 3.000
>
```

Figure 14: An example of the output from the ver command

5.2 h

The h command sets all subsequent calculations to use the 2-D horizontal mode of operation for the SAA. The SAAXYZ responds with the following message after the h command is sent.

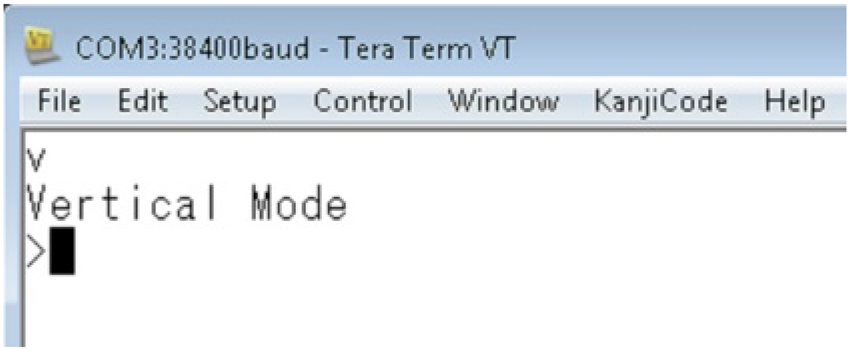

COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
h
Horizontal Mode
>
```

Figure 15: An example of the output from the h command

5.3 v

The v command sets all subsequent calculations to use the 3-D vertical mode of operation for the SAA. The SAAXYZ responds with the following message after the v command is sent.

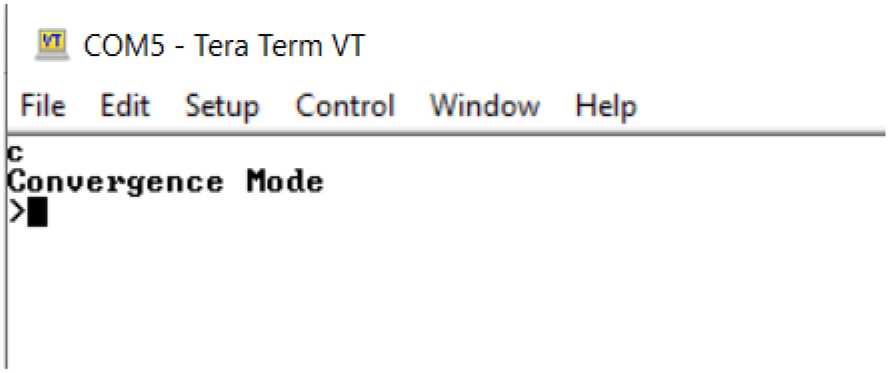

COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
v
Vertical Mode
>
```

Figure 16: An example of the output from the v command

5.4 c

The c command sets all subsequent calculations to use the 2-D convergence mode of operation for the SAA. The SAAXYZ responds with the following message after the c command is sent.

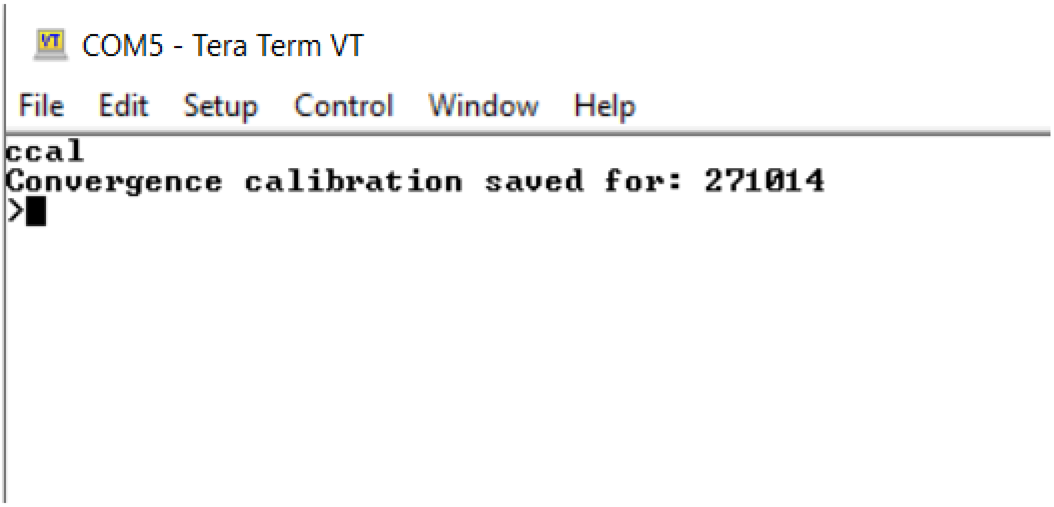


```
c
Convergence Mode
>
```

Figure 17:

5.5 ccal

The ccal command does a 2-D convergence calibration on all connected ShapeArrays. It requires that the convergence mode (see the 'c' command above) be set. The ccal command should be executed once after the convergence SAAs have been installed, and before any position measurements are made with either the 'pos' or 'post' command. It updates the calibration information stored for the SAA(s) in the SAAXYZ's memory.

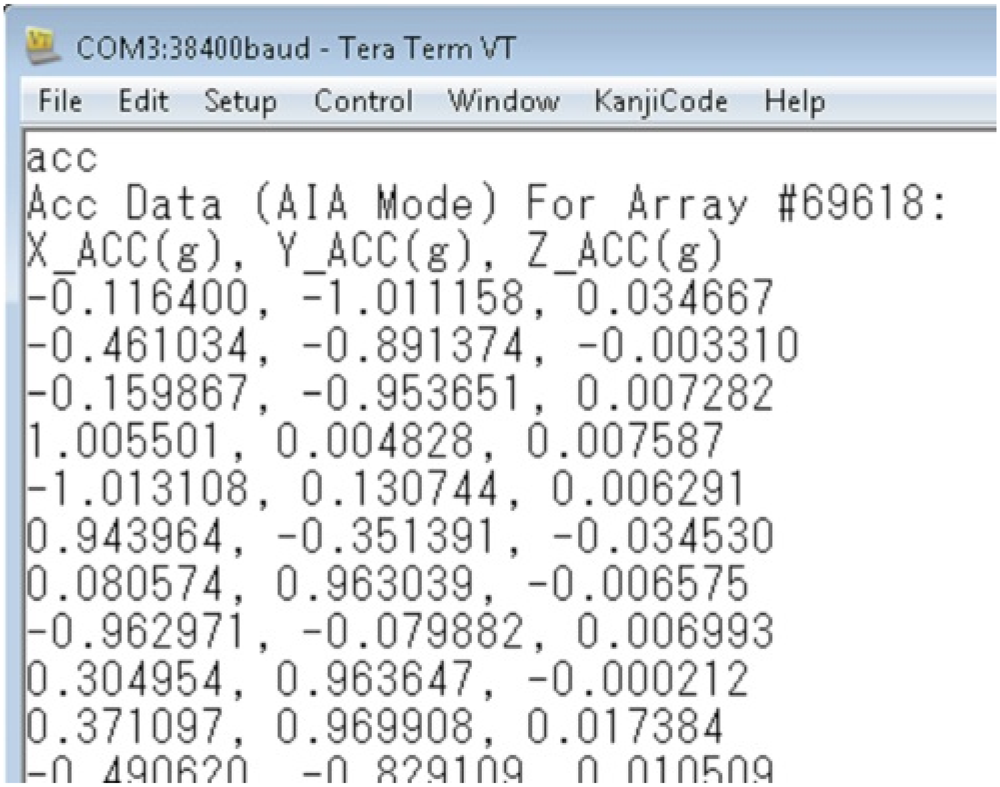


```
ccal
Convergence calibration saved for: 271014
>
```

Figure 18:

5.6 acc

The acc command requests acceleration data for all segments of all SAAs connected to the SAAXYZ. The raw data used to compute accelerations is averaged by the averaging level previously defined using the ‘avg’ command. The output from the SAAXYZ appears as follows.



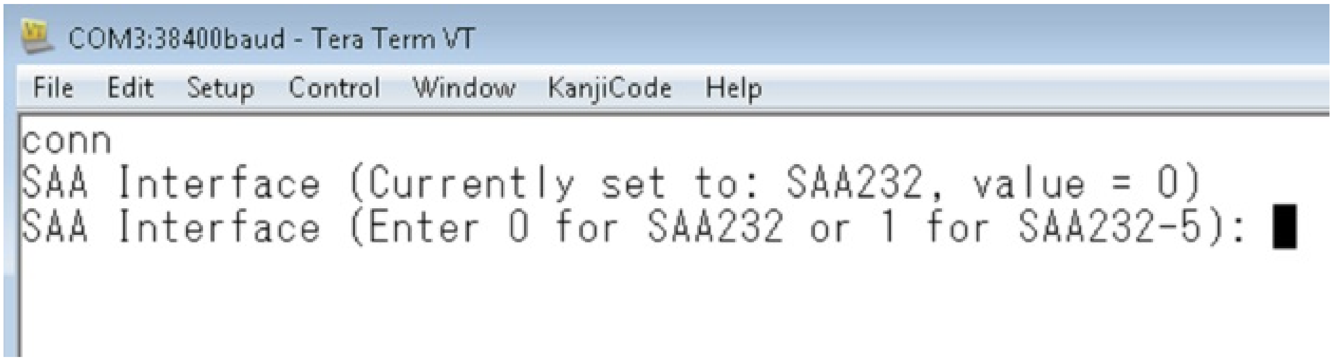
COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
acc
Acc Data (AIA Mode) For Array #69618:
X_ACC(g), Y_ACC(g), Z_ACC(g)
-0.116400, -1.011158, 0.034667
-0.461034, -0.891374, -0.003310
-0.159867, -0.953651, 0.007282
1.005501, 0.004828, 0.007587
-1.013108, 0.130744, 0.006291
0.943964, -0.351391, -0.034530
0.080574, 0.963039, -0.006575
-0.962971, -0.079882, 0.006993
0.304954, 0.963647, -0.000212
0.371097, 0.969908, 0.017384
-0.490620, -0.829109, 0.010509
```

Figure 19:

5.7 conn

The conn command prompts you to choose whether or not the SAAXYZ is connected to an SAA232 or an SAA232-5 device. The SAA232 is used when interfacing to just a single SAA. The SAA232-5 can be used with up to five SAAs. If the SAA232-5 is selected, you is prompted to enter or confirm the serial numbers of the SAAs that are connected to the SAA232-5 ports.



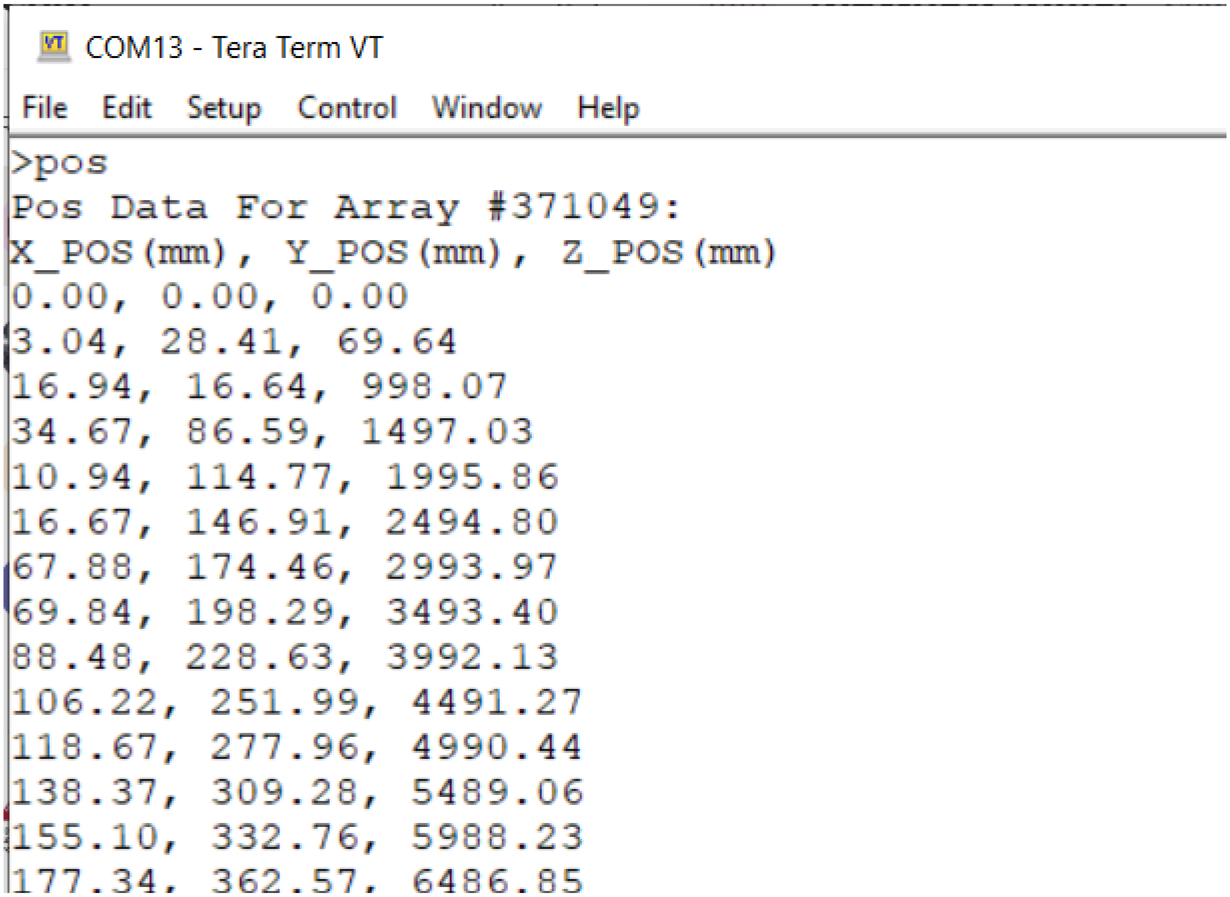
COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
conn
SAA Interface (Currently set to: SAA232, value = 0)
SAA Interface (Enter 0 for SAA232 or 1 for SAA232-5):
```

Figure 20:

5.8 pos

The pos command requests position data for all segments of connected SAAs. The raw data used to compute positions is averaged by the averaging level previously defined using the ‘avg’ command. The output from the SAAXYZ appears as follows.

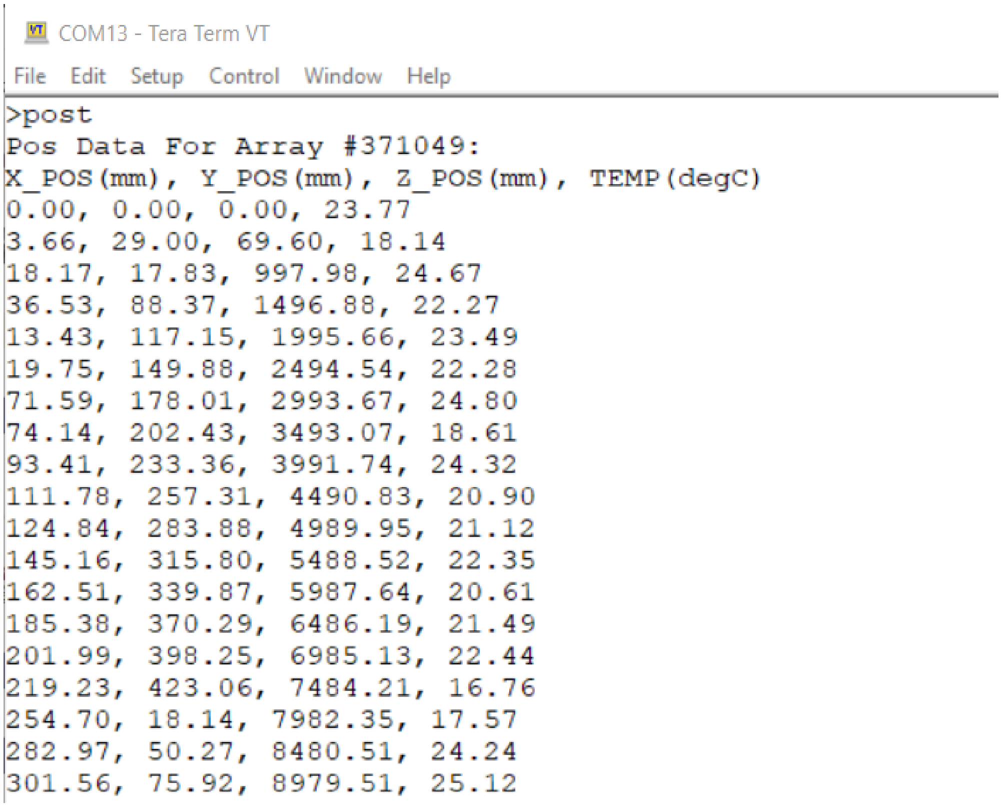


```
>pos
Pos Data For Array #371049:
X_POS(mm), Y_POS(mm), Z_POS(mm)
0.00, 0.00, 0.00
3.04, 28.41, 69.64
16.94, 16.64, 998.07
34.67, 86.59, 1497.03
10.94, 114.77, 1995.86
16.67, 146.91, 2494.80
67.88, 174.46, 2993.97
69.84, 198.29, 3493.40
88.48, 228.63, 3992.13
106.22, 251.99, 4491.27
118.67, 277.96, 4990.44
138.37, 309.28, 5489.06
155.10, 332.76, 5988.23
177.34, 362.57, 6486.85
```

Figure 21:

5.9 post

The post command requests position and temperature data for all segments of connected SAAs. The raw data used to compute positions is averaged by the averaging level previously defined using the ‘avg’ command. The output for the post command is similar to the output of the pos command shown in Figure 20 above; however, it also includes a TEMP(degC) column containing temperatures in degrees Celsius.

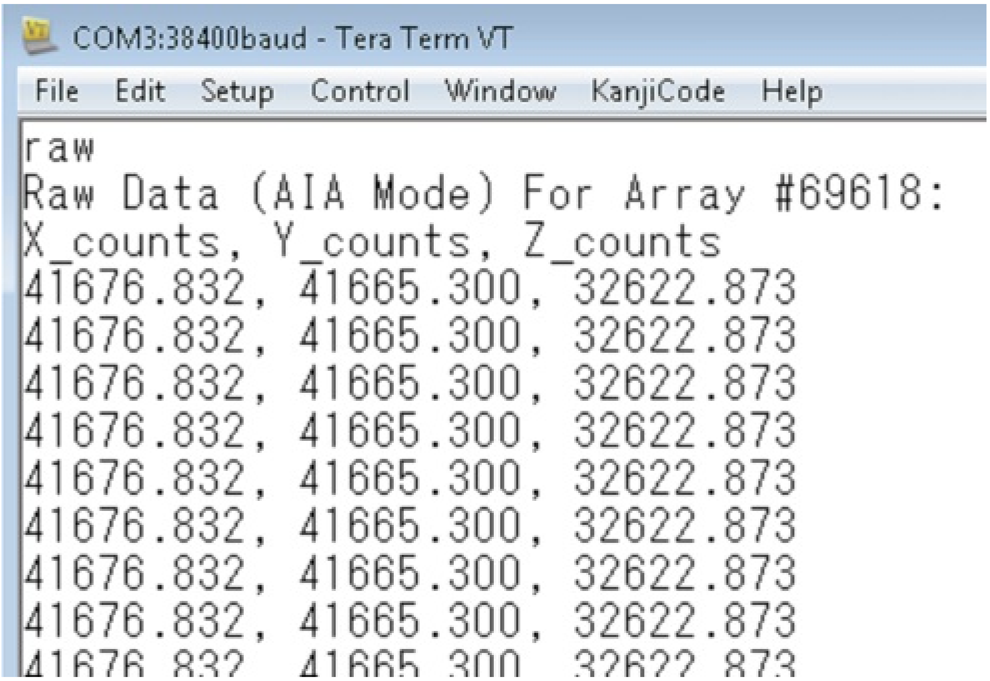


```
>post
Pos Data For Array #371049:
X_POS(mm), Y_POS(mm), Z_POS(mm), TEMP(degC)
0.00, 0.00, 0.00, 23.77
3.66, 29.00, 69.60, 18.14
18.17, 17.83, 997.98, 24.67
36.53, 88.37, 1496.88, 22.27
13.43, 117.15, 1995.66, 23.49
19.75, 149.88, 2494.54, 22.28
71.59, 178.01, 2993.67, 24.80
74.14, 202.43, 3493.07, 18.61
93.41, 233.36, 3991.74, 24.32
111.78, 257.31, 4490.83, 20.90
124.84, 283.88, 4989.95, 21.12
145.16, 315.80, 5488.52, 22.35
162.51, 339.87, 5987.64, 20.61
185.38, 370.29, 6486.19, 21.49
201.99, 398.25, 6985.13, 22.44
219.23, 423.06, 7484.21, 16.76
254.70, 18.14, 7982.35, 17.57
282.97, 50.27, 8480.51, 24.24
301.56, 75.92, 8979.51, 25.12
```

Figure 22:

5.10 raw

The raw command requests raw data from all segments of connected SAAs. The raw data is averaged by the averaging level previously defined using the ‘avg’ command. The output from the SAAXYZ appears as follows (N is the number of SAA segments).

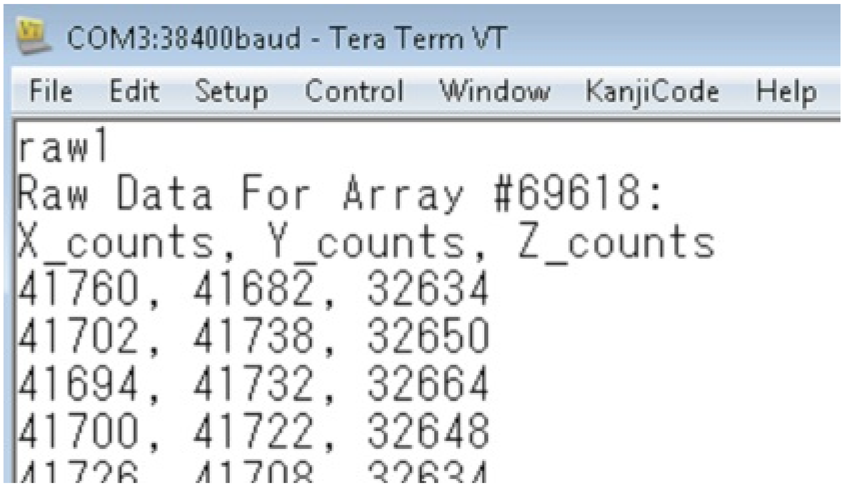

COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
raw
Raw Data (AIA Mode) For Array #69618:
X_counts, Y_counts, Z_counts
41676.832, 41665.300, 32622.873
41676.832, 41665.300, 32622.873
41676.832, 41665.300, 32622.873
41676.832, 41665.300, 32622.873
41676.832, 41665.300, 32622.873
41676.832, 41665.300, 32622.873
41676.832, 41665.300, 32622.873
41676.832, 41665.300, 32622.873
41676.832, 41665.300, 32622.873
```

Figure 23:

5.11 raw1

The raw1 command gets a single sample of raw data from all connected SAAs. The output from the SAAXYZ appears as follows.

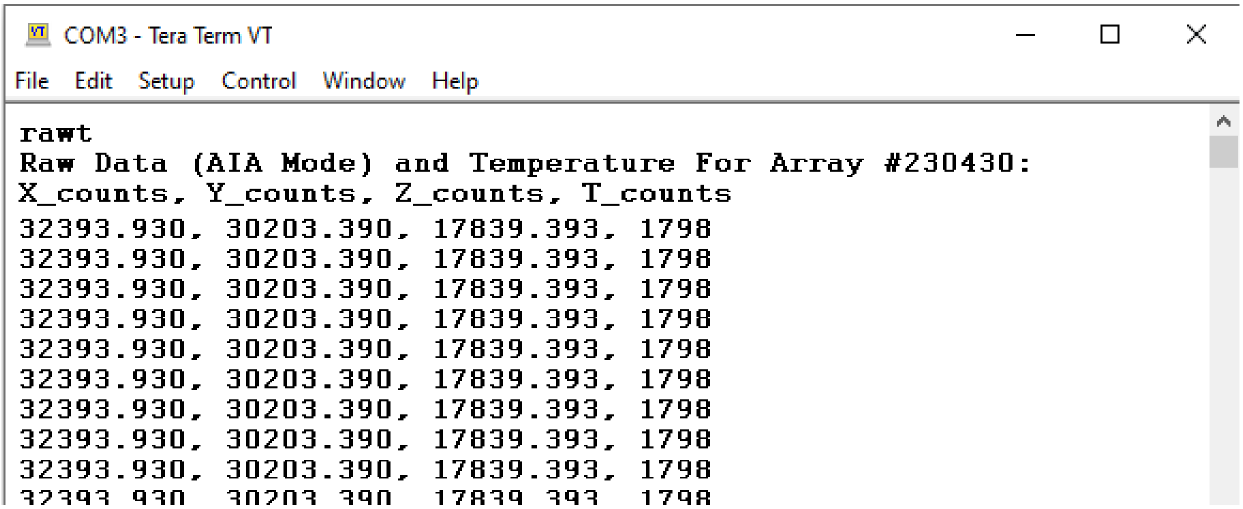

COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
raw1
Raw Data For Array #69618:
X_counts, Y_counts, Z_counts
41760, 41682, 32634
41702, 41738, 32650
41694, 41732, 32664
41700, 41722, 32648
41726, 41708, 32634
```

Figure 24:

5.12 rawt

The rawt command is intended for getting raw accelerometer sensor values and temperature data at the same time. It is similar to the "raw" command except temperature data is also included. This command is only supported for ShapeArrays with a serial number greater than or equal to 66000. For ShapeArrays with serial numbers between 66000 and 199999, the temperature is output in degrees Celsius. For ShapeArrays with serial numbers greater than or equal to 200000, the temperature is output as a value in raw counts which must be scaled by linear coefficients to obtain the temperature in degrees Celsius. For model 1 or model 2 ShapeArrays (i.e. serial numbers less than 66000) the response to this command is just an empty line.



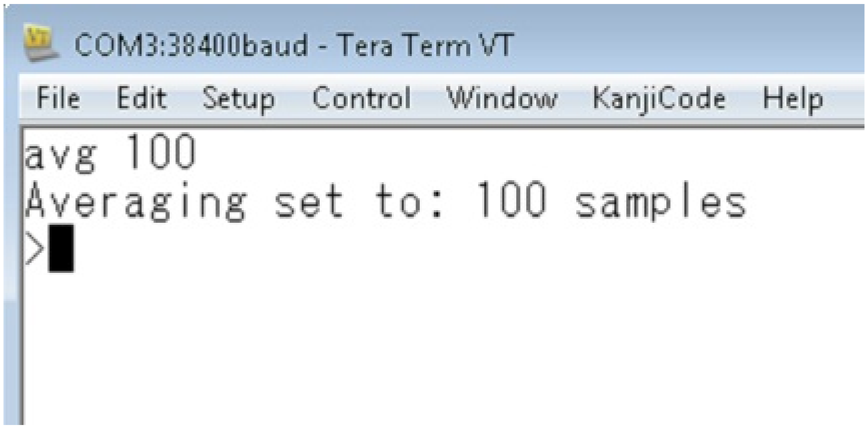
```
rawt
Raw Data (AIA Mode) and Temperature For Array #230430:
X_counts, Y_counts, Z_counts, T_counts
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
32393.930, 30203.390, 17839.393, 1798
```

Figure 25:

5.13 avg <*#_of_samples_to_average*>

The avg command sets the averaging level of the SAAXYZ for all subsequent SAA calculations. The averaging level must be set to a value between 100 and 25500, and must be a multiple of 100. The SAAXYZ responds with the following if averaging is set successfully:

Averaging set to: XXXX samples
XXXX is a number between 100 and 25500.


COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

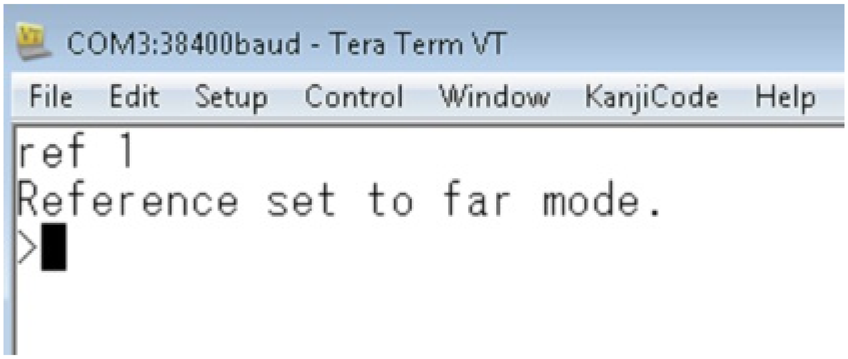
```
avg 100
Averaging set to: 100 samples
>
```

Figure 26:

5.14 ref <*0 or 1*>

The ref command sets the reference end of the SAA for subsequent calculations.

- "**ref 0**" to set the reference to be at the near (cable) end of the SAA
- "**ref 1**" to set the reference to be at the far (tip) end of the SAA.

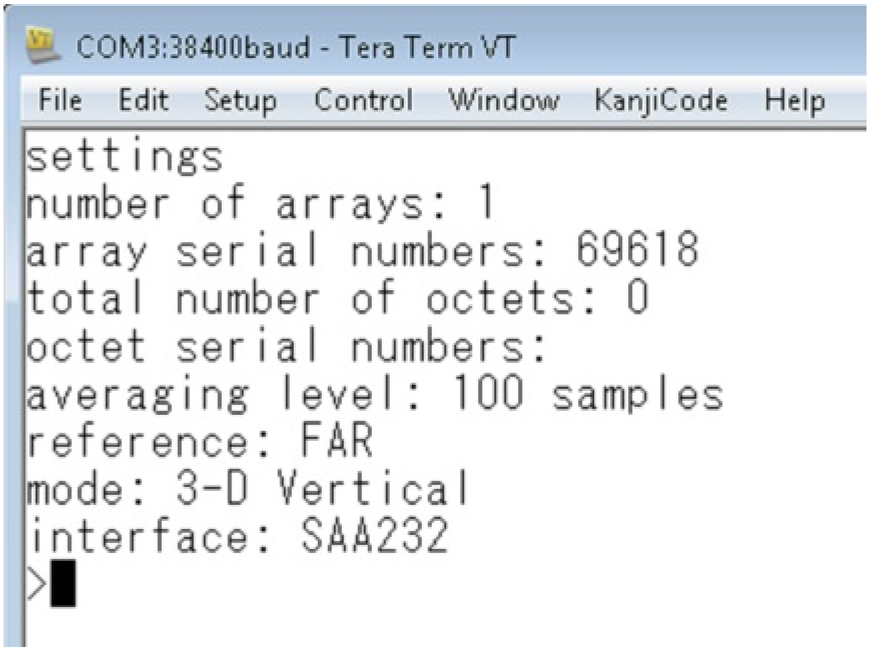

COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
ref 1
Reference set to far mode.
>
```

Figure 27: An example of the ref command setting the reference end to the far (tip) end of the SAA

5.15 settings

The settings command displays a list of the settings that are currently being used on the SAAXYZ. An example of these settings appears below.

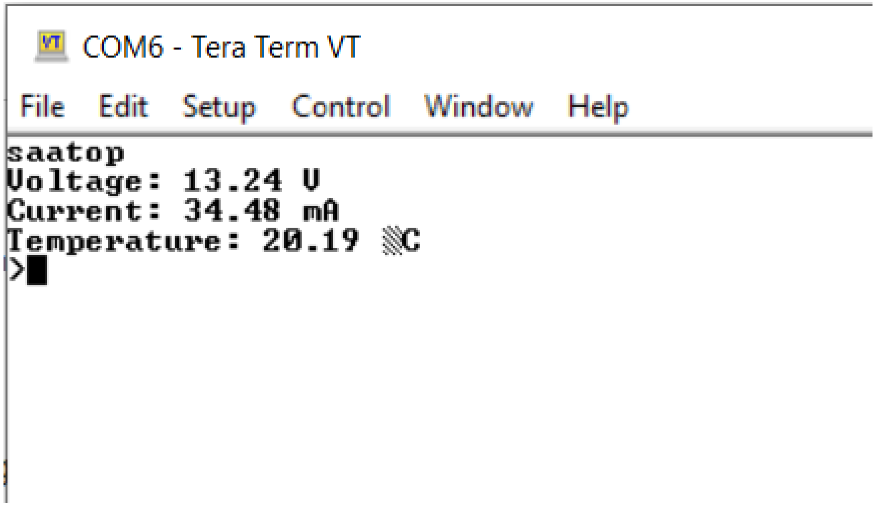

COM3:38400baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
settings
number of arrays: 1
array serial numbers: 69618
total number of octets: 0
octet serial numbers:
averaging level: 100 samples
reference: FAR
mode: 3-D Vertical
interface: SAA232
>
```

Figure 28:

5.16 saatop

The saatop command queries the top segment of the ShapeArray for measurements of the voltage (V), electrical current (mA), and temperature (° C) at the top of the instrument.



```
saatop
Voltage: 13.24 V
Current: 34.48 mA
Temperature: 20.19 ▒C
>
```

Figure 29:

6 Binary Protocol

This section describes the binary communications protocols that are used for controlling, retrieving status information, and getting data from Measurand's SAAXYZ device. (NOTE: the binary communications protocols do not support operation in convergence mode.)

When communicating with the SAAXYZ in binary mode, both the commands sent to the SAAXYZ, and the response(s) received from the SAAXYZ are expressed as hexadecimal packet strings. Each packet string sent to or received from the SAAXYZ must be of the following form:

```
:<packet length (2 bytes)><transaction ID byte><command byte> <data byte(s)><CRC-08 byte><0x0d byte><0x0a byte>
```

Each component of the packet is defined below:

1. Each packet must begin with a ":" character.
2. Two size bytes that indicate the length (in bytes) of the packet remaining after these two size bytes:
   - Both of these bytes are expressed as two hexadecimal characters, so that there are four hexadecimal characters in total representing the remaining packet length
   - For example: 001C would mean that there are 28 bytes after the size bytes in the rest of the packet
3. Transaction ID byte is always set to 01 for now
4. Command Byte ranges from 01 to FF (expressed as two hexadecimal characters) and corresponds to a particular command for the SAAXYZ.
5. Data Bytes:
   - These bytes contain information corresponding to the particular command byte that precedes them in the packet
   - They are also expressed as hexadecimal bytes (two hexadecimal characters per byte), for example: 011A21 represents the three bytes expressed decimally as: 1, 26, 33
6. CRC-08 Byte:
   - This byte is formed by computing the CRC-08 checksum of the ":" character, the four hexadecimal packet size characters, the two transaction ID characters, the two command byte characters, and any data characters, in that order
   - The CRC-08 byte is also expressed as two hexadecimal characters, corresponding to a byte value between 0x00 and 0xff
7. 0x0d and 0x0a Bytes:

Each packet must end with a carriage return, linefeed sequence, i.e. 0x0d, 0x0a

The following example gets a list of octet serial numbers connected to the SAAXYZ:

```
:00080108D4<cr><lf>
```

The SAAXYZ would then respond to the above command packet with a response packet similar to the following:

Hexadecimal Value	Command Description
01	Get averaging level used by SAAXYZ.
02	Get mode (2D or 3D) used for SAAXYZ calculations.
03	Get reference end (near or far) used for SAAXYZ calculations.
04	Set Averaging Level used by SAAXYZ.
05	Set Mode (2D or 3D) used for SAAXYZ calculations.
06	Set reference end (near or far) used for SAAXYZ calculations.
07	Get total # of octets (models 1 and 2 SAAs only) connected to the SAAXYZ.
08	Get list of octet serial numbers (models 1 and 2 SAAs only) connected to the SAAXYZ.
09	Get raw data from an individual octet (models 1 and 2 SAAs only).
0A	Error information packet.
0B	Acquire raw data for all available SAAs in AIA mode.
0C	Get serial numbers of all SAAXYZ SAAs.
0D	Get number of octets in a particular model 1 or model 2 SAA connected to the SAAXYZ, and also returns the octet serial numbers of that SAA.
0E	Get raw data from a particular model 1 or model 2 SAA connected to the SAAXYZ.
0F	Get acceleration data from a particular segment of a model 1 or model 2 SAA connected to the SAAXYZ.
10	Get acceleration data from an entire octet of a model 1 or model 2 SAA.
11	Get acceleration data from an entire (model 1 or model 2) SAA.
12	Get position data for a particular joint of a model 1 or model 2 SAA.
13	Get total number of SAAs connected to the SAAXYZ.

Hexadecimal Value	Command Description
14	Get position data from an entire octet of a model 1 or model 2 SAA.
15	Get position data from an entire model 1 or model 2 SAA.
16	Get temperature data from an entire octet of a model 1 or model 2 SAA.
17	Get temperature data from an entire model 1 or model 2 SAA.
18	Set Baud rate for communications between the SAAXYZ and the computer or other host that is connected to it.
19	Get total number of model 3 (or higher) segments that are available on the SAAXYZ.
1A	Get number of model 3 (or higher) segments that are available in a particular SAA connected to the SAAXYZ.
1B	Get raw data for a particular model 3 (or higher) SAA connected to the SAAXYZ.
1C	Get raw data for an individual segment of a model 3 (or higher) SAA.
1D	Get acceleration data for an individual segment of a model 3 (or higher) SAA.
1E	Get acceleration data for an entire model 3 (or higher) SAA.
1F	Get position data for an individual segment of a model 3 (or higher) SAA
20	Get position data for an entire model 3 (or higher) SAA.
21	Get temperature data for an entire model 3 (or higher) SAA.

Table 1: Binary Protocol Commands

See below for a description of the command to get octet serial numbers. This example response indicates that 3 octets are available with the serial numbers 0xB93D, 0xB93F, and 0xB940; i.e. 47421, 47423, and 47424 in decimal notation.

7 SAAXYZ Binary Command Reference

7.1 Averaging Level (Command Byte 0x01)

Send the following bytes to the SAAXYZ to determine the number of samples that it is using for averaging:

```
:0008010196<cr><lf>
```

The averaging level returned from the SAAXYZ should range between 100 and 25500. For example, if the averaging level is 1000, the SAAXYZ will respond with the following bytes:

```
:000C010103E840<cr><lf>
```

7.2 Get Mode (2D or 3D) (Command Byte 0x02)

Send the following bytes to the SAAXYZ to determine the mode (2D or 3D) that is being used for SAA calculations:

```
:00080102DA<CR><LF>
```

The mode returned from the SAAXYZ should be either 0 or 1. If it is 0, that means that the SAAXYZ is in 3D mode (i.e. calculates SAA shape in three dimensions, X, Y, and Z). If it is 1, that means that the SAAXYZ is in 2D mode (i.e. calculates SAA shape in two dimensions, X and Z).

For example, if the SAAXYZ is currently in 2D mode, it will respond with the following packet:

```
:000A010201DA<CR><LF>
```

7.3 Get Reference End (Command Byte 0x03)

Send the following bytes to the SAAXYZ to determine the reference end (i.e. near or far) of the SAA(s):

```
:000801037C<cr><lf>
```

If the data byte in the returned packet is 0x00, then the reference end is the near (cable) end of the SAA. If the data byte in the returned packet is 0x01, then the reference end is the far end of the SAA.

For example, if the SAAs are being referenced to the near end, then the SAAXYZ will respond with the following packet:

```
:000A01030034<cr><lf>
```

7.4 Set Averaging Level (Command Byte 0x04)

This command byte is used to set the averaging level used for SAAXYZ calculations. The averaging level must be set to be between 100 and 25500 samples. The averaging level is represented as a two byte (four hexadecimal character) value, with the most significant byte followed by the least significant byte.

For example, to set the averaging level to be 1000, send the following packet to the SAAXYZ:

```
:000C010403E84C<cr><lf>
```

7.5 Set Mode (2D or 3D) (Command Byte 0x05)

This command byte is used to set the mode (2D or 3D) used for SAAXYZ calculations.

For 3D mode, send 00 in the data section of the packet. For 2D mode, send 01 in the data section of the packet. For example, to set the SAAXYZ to use 2D mode, send the following packet to the SAAXYZ:

```
:000A01050184<cr><lf>
```

7.6 Set Reference End (Near or Far) (Command Byte 0x06)

This command byte is used to set the reference end (near or far) used for SAAXYZ calculations. To make the near (cable) end the reference, send 00 in the data section of the packet. To make the far end the reference, send 01 in the data section of the packet.

For example, to set the SAAXYZ to use the near end of the SAA(s) as the reference, send the following packet to the SAAXYZ:

```
:000A010600FA<cr><lf>
```

7.7 Get Total Number of Octets Connected to SAAXYZ (Command Byte 0x07)

This command byte is used to get the total number of SAA octets (i.e. groupings of 8 SAA segments) that are connected to the SAAXYZ. The command packet appears as follows:

```
:000801070E<cr><lf>
```

The SAAXYZ responds with a packet containing four hexadecimal characters in the data portion of the packet that represent the total number of SAA octets connected to the SAAXYZ. For example, if a total of three octets are connected, the SAAXYZ will respond with the following packet:

```
:000C0107000370<cr><lf>
```

7.8 Get List of Octet Serial Numbers (Command Byte 0x08)

This command byte is used to get a list of the serial numbers of all of the octets that are connected to the SAAXYZ. The command packet appears as follows:

```
:00080108D4<cr><lf>
```

The SAAXYZ responds with a packet containing the same command byte (0x08) and data bytes including four hex characters indicating the total number of octets, followed by four hex characters for the serial numbers of each of the octets. For example, for 3 octets with serial numbers of 47421, 47423, and 47424 the response from the SAAXYZ will be:

```
:001801080003B93DB93FB940BA<cr><lf>
```

When two or more SAAs are present, the octet serial numbers are returned firstly for the first SAA, then for the second SAA, the third SAA, and so on.

For example, if two SAA serial numbers are present, SAA #1 (A) has 3 octets and SAA #2 (B) has 5 octets, then the octet serial numbers will be returned in the order: A1, A2, A3, B1, B2, B3, B4, B5.

7.9 Get Raw Data from an Individual Octet (Command Byte 0x09)

This command byte is used to get raw data from an individual SAA octet that is connected to the SAAXYZ. The command packet appears as follows:

```
:000C0109B93D90<cr><lf>
```

The SAAXYZ responds with a packet containing the same command byte (0x09) and data bytes that are comprised of the floating point X,Y,Z raw data values packed together in X,Y,Z near to far order (i.e. X1, Y1, Z1, X2, Y2, Z2, ..., X8, Y8, Z8). The data bytes are followed by the usual CRC-08 byte and 0x0d, 0x0a terminator bytes.

The first 9 bytes of the response from the SAAXYZ should be as follows:

```
:00C80109
```

NOTE: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.10 Error Information Packet

The error information packet is not a command packet, but rather is an error response that the SAAXYZ will give instead of a normal response packet if an error happens to occur.

The first 9 bytes of the error information packet will be as follows:

```
:000C010A
```

The next 4 characters of the response indicate the possible error codes, which are as follows:

Error Code	Description
0001	Raw data has not been received yet. This error is typically generated if a user requests raw, acceleration, or position data values but has not yet instructed the SAAXYZ to actually acquire an AIA data sample first (by sending a 0x0B command byte to it).
0002	The octet referenced in the command sent to the SAAXYZ is not included in the device's list of available octets.
0003	There was an error trying to communicate with one or more SAAs.
0004	There was a CRC-08 checksum error in the last command sent to the SAAXYZ.
0005	Command sent to the SAAXYZ did not have the proper 0x0d, 0x0a terminator bytes at the end of it.
0006	The SAA serial number referenced in the last command to the SAAXYZ was invalid.
0007	The SAA segment number referenced in the last command sent to the SAAXYZ was invalid.

Error Code	Description
0008	The octet serial number referenced in the last command sent to the SAAXYZ was invalid.
0009	The baud rate specified for SAAXYZ communications with the host is not valid.
000A	Insufficient memory to provide a valid response.

Table 2: Error Codes

7.11 Acquire Raw Data for All Available Octets in AIA Mode (Command Byte 0x0B)

This command byte should be sent to the SAAXYZ to tell it to collect a sample of AIA (averaging-in-array) data from all available octets that are currently connected to it. This command packet should appear as follows:

```
:0008010B76<cr><lf>
```

After receiving this command, the SAAXYZ goes and collects the SAA data, and then responds with a confirmation packet, which is the same as the initial request packet:

```
:0008010B76<cr><lf>
```

> **NOTE**: Depending on the averaging level set previously using the 0x04 command byte (see above), you will need to wait a length of time (in seconds) at least equal to the averaging level divided by 400 plus about 1 additional second before receiving the confirmation packet from the SAAXYZ that the SAA data was collected. For example, if the current averaging level were set to 1000 samples, then you should wait at least 1000 / 400 + 1 = 3.5 seconds before expecting to see a return confirmation packet from the SAAXYZ that the data acquisition has taken place.

7.12 Get Serial Numbers of All SAAXYZ SAAs (Command Byte 0x0C)

This command returns a list of all of the SAA serial numbers presently available on the SAAXYZ. The command packet appears as follows:

```
:0008010CD0<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x0C command byte, as well as data bytes indicating the number of SAAs available (four hex characters), followed by the serial numbers of each of those SAAs (another four hex characters for each SAA).

For example, if a single SAA with serial number 47421 were available, the SAAXYZ would respond with the following packet:

```
:0010010C0001B93DB8<cr><lf>
```

7.13 Get Number of Octet Serial Numbers in a Particular SAAXYZ SAA (Command Byte 0x0D)

This command returns the number of octets present in a given SAA, as well as the octet serial numbers that are present within that SAA.

For example, to get the number of octets and octet serial numbers for the SAA with serial number 50658, the following packet would be sent to the SAAXYZ:

```
:000C010DC5E21E<cr><lf>
```

The SAAXYZ responds with a packet containing the number of octets in the specified SAA, expressed as a four hex character value, and followed by a four hex character value for each octet serial number. For example, if the SAA had 8 octets with the serial numbers: 50658, 50660, 50661, 50673, 50675, 50999, 51000, 51002; it would respond with the following:

```
:002C010D0008C5E2C5E4C5E5C5F1C5F3C737C738C73A4C<cr><lf>
```

7.14 Get Raw Data from a Particular SAAXYZ SAA (Command Byte 0x0E)

This command is used to request raw data from a particular SAAXYZ SAA. The data section of this command specifies the serial number of the SAA as a four byte hex number.

For example, to request raw data for the SAA with serial number 47421, the following packet should be sent to the SAAXYZ:

```
:000C010EB93DA2<cr><lf>
```

The SAAXYZ responds with a “Get Raw Data From an Individual Octet” packet (with a 0x09 command byte) for each octet available on the SAA. For this reason, it is recommended that you know how many octets are present on the SAA before calling this command (i.e. should call the “Get Number of Octet Serial Numbers in a Particular SAAXYZ SAA” command (0x0D command byte) at least once prior to calling this command.

> **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.15 Get Acceleration Data from a Particular SAA Segment (Command Byte 0x0F)

This command is used to acquire three axes of acceleration data (X, Y, and Z) from a specific segment of the SAA. The data bytes contain a 4 hex byte description of the SAA serial number, followed by a 4 hex byte description of the SAA segment number (relative to the reference end of the SAA).

For example, to get the acceleration of the 2nd segment of the SAA with serial number 47421, the following packet should be sent:

```
:0010010FB93D0002A2<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x0F command byte, followed by 12 characters of hex data that represent the three 4-byte floating point values for the X, Y, and Z axes accelerations of the specified segment.

> **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.16 Acceleration Data from an Entire Octet of an SAA (Command Byte 0x10)

This command is used to acquire acceleration data from an entire octet of the SAA. The data bytes contain a 4 hex byte description of the octet serial number.

For example, to obtain acceleration data from octet 47423, the following command packet should be sent to the SAAXYZ:

```
:000C0110B93F48<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x10 command byte, followed by 192 characters of hex data that represent the 24 4-byte floating-point values for the X, Y, and Z axes accelerations of all segments of the octet. The acceleration values are ordered in the returned packet as follows:

```
Acc.X1, Acc.Y1, Acc.Z1, Acc.X2,Acc.Y2,Acc.Z2, ... , Acc.X8,Acc.Y8,Acc.Z8
```

Where Acc.X1 corresponds to the X-axis acceleration of the segment closest to the near end, Acc.Y1 corresponds to the Y-axis acceleration of the segment closest to the near end, etc.

> ⓘ **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.17 Get Acceleration Data from an Entire SAA (Command Byte 0x11)

This command is used to acquire acceleration data from an entire SAA. The data bytes contain a 4 hex byte description of the SAA serial number.
For example, to obtain acceleration data from the SAA 47421, the following command packet should be sent to the SAAXYZ:

```
:000C0111B93D3C<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x11 command byte, followed by N x 8 x 3 x 4 x 2 = N x 192 characters of hex data that represent the 24 4-byte floating point values for all segments of each octet of the SAA. (N = the total number of octets in the SAA.) The values are presented in the following order:

```
Acc.X1, Acc.Y1, Acc.Z1, Acc.X2,Acc.Y2,Acc.Z2, etc ...
```

Where Acc.X1 corresponds to the X-axis acceleration of the segment closest to the reference end, Acc.Y1 corresponds to the Y-axis acceleration of the segment closest to the reference end, etc.

> ⓘ **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.18 Get Position Data for a Particular Joint of an SAA (Command Byte 0x12)

This command is used to acquire three axes of position data (X, Y, and Z) from a specific joint of the SAA. The data bytes contain a 4 hex byte description of the SAA serial number, followed by a 4 hex byte description of the SAA joint number (relative to the reference end of the SAA).

For example, to get the position of the 2nd joint of the SAA with serial number 47421, the following packet should be sent:

```
:00100112B93D00025C<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x12 command byte, followed by 12 characters of hex data that represent the three 4-byte floating-point values for the X, Y, and Z axes accelerations of the specified segment.

> **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.19 Get Total Number of SAAs Connected to the SAAXYZ (Command Byte 0x13)

This command is used to find out how many SAAs are available on the SAAXYZ. The following bytes should be sent to the SAAXYZ:

```
:0008011304<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x13 command byte, followed by 4 characters of hex data that represent the total number of SAAs available. For example, if just one SAA is available, it will respond with the following packet:

```
:000C0113000126<cr><lf>
```

> **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.20 Get Position Data from an Entire Octet of an SAA (Command Byte 0x14)

This command is used to acquire position data from an entire octet of the SAA. The data bytes contain a 4 hex byte description of the octet serial number.

For example, to obtain position data from octet 47423, the following command packet should be sent to the SAAXYZ:

```
:000C0114B93FDA<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x14 command byte, followed by 216 characters of hex data that represent the 27 4-byte floating-point values for the X, Y, and Z axes positions of all joints of the octet. The position values are ordered in the returned packet as follows:

```
Pos.X0,Pos.Y0,Pos.Z0, Pos.X1,Pos.Y1,Pos.Z1, ... , Pos.X8,Pos.Y8,Pos.Z8
```

Where 0 corresponds to the joint of the octet closest to the reference end, and 8 corresponds to the joint of the octet furthest from the reference end.

NOTE: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.21 Get Position Data from an Entire SAA (Command Byte 0x15)

This command is used to acquire position data from an entire SAA. The data bytes contain a 4 hex byte description of the SAA serial number.

For example, to obtain position data from the SAA 47421, the following command packet should be sent to the SAAXYZ:

```
:000C0115B93DAE<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x15 command byte, followed by (N x 8 + 1) x 3 x 4 x 2 = (N x 8 + 1) x 24 characters of hex data that represent the 24 4-byte floating-point values for all joints of each octet of the SAA, plus 3 final floating-point values for the very end of the SAA. (N = the total number of octets in the array.) The values are presented in the following order:

```
Pos.X0, Pos.Y0, Pos.Z0, Pos.X1,Pos.Y1,Pos.Z1, etc ...
```

Where Pos.X0 corresponds to the X-axis position of the joint at the reference end, Pos.Y0 corresponds to the Y-axis position of the joint at the reference end, etc.

NOTE: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.22 Get Temperature Data from an Entire Octet of an SAA (Command Byte 0x16)

This command is used to acquire temperature data from an entire octet of the SAA. The data bytes contain a 4 hex byte description of the octet serial number.

For example, to obtain temperature data from octet 47423, the following command packet should be sent to the SAAXYZ:

```
:000C0116B93F40<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x16 command byte, followed by 4 characters of hex data that represent the 4-byte floating-point value for the temperature of the octet (each octet has a single temperature sensor).

> ⓘ **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.23 Get Temperature Data from an Entire SAA (Command Byte 0x17)

This command is used to acquire temperature data from an entire SAA. The data bytes contain a 4 hex byte description of the SAA serial number.
For example, to obtain temperature data from the SAA 47421, the following command packet should be sent to the SAAXYZ:

```
:000C0117B93D34<cr><lf>
```

The SAAXYZ responds with a packet containing the 0x17 command byte, followed by N x 4 x 2 = N x 8 characters of hex data that represent the floating-point temperature values for each octet of the array (each octet has its own temperature sensor). The octet temperatures are returned in order starting at the reference end of the SAA.

> ⓘ **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.24 Set Baud Rate For Communications Between SAAXYZ and Host (Command Byte 0x18)

This command can be used to change the baud rate that is used for communications between the host computer and the SAAXYZ. The data portion of the packet sent to the SAAXYZ contains a 4-byte value for the desired baud rate.

For example, to set the baud rate to 38400 bps the following command should be sent:

```
:001001180000960022<cr><lf>
```

Or to set the baud rate to 115200 bps, the following command should be sent:

```
:001001180001C20046<cr><lf>
```

Valid possible baud rates include 9600 bps, 19200 bps, 38400 bps, 57600 bps, and 115200 bps.

7.25 Get Total Number of Model 3 (or higher) Segments Available on the SAAXYZ (Command Byte 0x19)

This command is used to determine the total number of model 3 (or higher) segments available on an SAAXYZ. The command packet is as follows:

```
:000801190A<cr><lf>
```

The data part of the response is a two-byte value indicating the total number of model 3 segments present in the SAAXYZ. For example, if a total of 231 model 3 segments are available, the SAAXYZ would respond as follows:

```
:000C011900E7EE<cr><lf>
```

7.26 Get Number of Model 3 (or higher) Segments Available in an SAA connected to the SAAXYZ (Command Byte 0x1A)

This command is used to determine the number of segments that a particular model 3 (or higher) SAA has. The data part of the command request packet contains three bytes to specify the serial number of the SAA.

For example, to request the number of segments present in an SAA with serial number 69618:

```
:000E011A010FF27E<cr><lf>
```

The response contains a two-byte data part indicating how many segments are present in the SAA. For example, the SAA #69618 contains 200 segments:

```
:000C011A00C822<cr><lf>
```

7.27 Get Raw Data for a Model 3 (or higher) SAA connected to the SAAXYZ (Command Byte 0x1B)

This command is used to get raw data values from a model 3 (or higher) SAA. The data part of the request packet uses three bytes to specify the serial number of the SAA.

For example, to request raw data from the SAA with serial number 69618, the following request packet would be used:

```
:000E011B010FF238<cr><lf>
```

The SAAXYZ responds with a separate raw data packet for each segment. Each raw data packet in the response uses the command byte 0x1C and 12 bytes of raw data for the segment in X, Y, Z order, floating-point format.

> **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.28 Receive Raw Data for an Individual Segment of a Model 3 (or higher) SAA (Command Byte 0x1C)

To receive raw data values from a model 3 (or higher) SAA, the 0x1B command byte (see above) should be used. The SAAXYZ then responds with an individual 0x1C packet for each segment of the SAA, which contains 12 bytes of floating-point raw data for each segment in X, Y, Z order.

> **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.29 Get Acceleration Data for an Individual Segment of a Model 3 (or higher) SAA (Command Byte 0x1D)

This command is used to get acceleration data (in G) for an individual segment of a model 3 (or higher) SAA. The data part of the request packet uses three bytes to specify the serial number of the SAA, followed by two bytes to specify the segment number (relative to the reference end) of the SAA. The acceleration is returned in a packet with the same 0x1D command byte, and 12 bytes of floating-point data, specifying the acceleration in X, Y, Z order of the requested segment.

For example, to get the acceleration of the second segment of the SAA #69618, you would send the following packet:

```
:0012011D010FF200021C<cr><lf>
```

An example of a possible response would be:

```
:0020011D7C0BD3BE2CBB68BF6CB9003D9E<cr><lf>
```

The above response packet corresponds to a segment acceleration of:

```
Acc_X = -0.4122 G
Acc_Y = -0.9091 G
Acc_Z = 0.0314 G
```

ⓘ **NOTE**: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.30 Get Acceleration Data for an Entire Model 3 (or higher) SAA (Command Byte 0x1E)

This command is used to get all of the segment acceleration values from an entire model 3 (or higher) SAA. The data part of the request packet contains three bytes to specify the serial number of the SAA. Floating-point acceleration values are then returned in the response packet for all of the SAA’s segments in the same packet. The values are returned in the order X1, Y1, Z1, X2, Y2, Z2, … XN, YN, ZN where N is the number of SAA segments.

For example, to request acceleration data for the entire SAA #69618, send the following command packet:

```
:000E011E010FF214<cr><lf>
```

NOTE: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.31 Get Position Data for an Individual Vertex of a Model 3 (or higher) SAA (Command Byte 0x1F)

This command is used to get X, Y, Z position values for a single model 3 (or higher) SAA vertex. The data part of the request packet contains three bytes to specify the serial number of the SAA and two bytes to specify the vertex number (relative to the reference end) of the SAA. Floating-point values are returned in X, Y, Z order for the requested vertex.

For example, to request position data for the second vertex of SAA #69618, send the following command packet:

```
:0012011F010FF20002CC<cr><lf>
```

NOTE: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

7.32 Get Position Data for an Entire Model 3 (or higher) SAA (Command Byte 0x20)

This command is used to get all of the vertex position values for an entire model 3 (or higher) SAA. The data part of the request packet contains three bytes to specify the serial number of the SAA. The response packet contains floating-point values for all of the SAA vertices, in the following order: X1, Y1, Z1, X2, Y2, Z2, … XN, YN, ZN, X(N+1), Y(N+1), Z(N+1), where N is the number of SAA segments.

For example, to request position data for the entire SAA #69618, send the following command packet:

```
:000E0120010FF272<cr><lf>
```

NOTE: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.

Get Temperature Data for an Entire Model 3 (or higher) SAA (Command Byte 0x21)

This command is used to get all of the segment temperature values for an entire model 3 (or higher) SAA. The data part of the request packet contains three bytes to specify the serial number of the SAA. The response packet contains a single floating point value for each SAA segment, in the following order: T1, T2, … TN, where N is the number of SAA segments.

For example, to request temperature data for the entire SAA #69618, send the following command packet:

```
:000E0121010FF2D2<cr><lf>
```

NOTE: The 0x0B command for acquiring a sample of AIA data from all available SAAs must be sent to the SAAXYZ prior to issuing this command in order to guarantee that current data is available for the SAAXYZ to use.