

API Reference

Table of Contents

	Target	Title	/get	/set
1	mmsp	Modular Mass Spec	✓	✓
2	electronicsInfo	Electronics Information	✓	✓
3	serialNumber	Serial Number	✓	✓
4	monitorFWVersion	Monitor Firmware Version	✓	X
5	controlFWVersion	Control Firmware Version	✓	X
6	genus	Genus	✓	✓
7	massRange	Mass Range	✓	X
8	communication	Communications Port Configuration	✓	✓
9	ipAddress	IP Address	✓	✓
10	macAddress	MMSP MAC Address	✓	✓
11	gateway	MMSP Gateway	✓	✓
12	mask	MMSP Mask	✓	✓
13	port	MMSP Port	✓	✓
14	dhcp	Boot from DHCP	✓	✓
15	sessionID	Session ID	✓	✓
16	issueLog	Issue Log	✓	✓
17	errorLog	Error Log	✓	X
18	sessionTimeout	Session Timeout	✓	✓
19	locationIdentifyCPU	Identify MMSP Location using the CPU Power LED	✓	✓
20	login	Login	✓	✓
21	clientName	Client Name	✓	✓
22	control	Control	✓	✓
23	controlInfo	Control Info	✓	X
24	sessionID		✓	✓
25	ipAddress		✓	✓
26	amInControl		✓	✓
27	canTakeControl		✓	✓
28	canForceControl		✓	✓
29	secondsSinceLastRequest		✓	✓
30	secondsSinceLastControlRequest		✓	✓
31	secondsSinceLastDataRequest		✓	✓
32	clientName		✓	✓
33	amInControl	Am In Control	✓	X

	Target	Title	/get	/set
34	controlLock	ControlLock	✓	✓
35	generalControl	General control	✓	✓
36	setEmission	Set Emission State	✓	✓
37	degas	Set Degas State	✓	✓
38	setEM	Set Electron Multiplier State	✓	✓
39	rfGeneratorSet	RF Generator Set	✓	✓
40	baselineMeasureSet	Baseline Measure Set	✓	✓
41	baselineCorrectSet	Baseline Correction Set	✓	✓
42	peakfindMeasureSet	Peakfind Measure State Set	✓	✓
43	peakfindCorrectSet	Peakfind Correct Set	✓	✓
44	emEquivIonSet	EM Equivalent Ion Current Set	✓	✓
45	ppLinearization	Partial Pressure Linearization	✓	✓
46	fanState	Fan State	✓	✓
47	userTimer1	User Timer 1	✓	✓
48	userTimer2	User Timer 2	✓	✓
49	shutdown	Shutdown	X	✓
50	sensorInfo	Sensor Information	✓	✓
51	name	Sensor Name	✓	✓
52	description	Sensor Description	✓	✓
53	serialNumber	Sensor Serial Number	✓	✓
54	ionSourceType	Ion Source Type	✓	✓
55	massFilterType	Ion Source Type	✓	✓
56	detectorType	Ion Source Type	✓	✓
57	filaments (array indexed by 0, maxItems: 2)		✓	✓
58	filamentType	Filament Type	✓	✓
59	status	Status	✓	✓
60	systemStatus	System Status	✓	X
61	systemStatus2	System Status 2	✓	X
62	hardwareErrors	Hardware errors	✓	✓
63	fil1Open	Filament 1 Open	✓	✓
64	fil2Open	Filament 2 Open	✓	✓
65	hardwareWarnings	Hardware warnings	✓	✓
66	powerSupplyPowerOnTime	Power Supply PowerOnTime	✓	✓
67	emissionStretch	EmissionStretch	✓	X
68	emStretch	EMStretch	✓	X
69	cpuPowerOnTime	CPU PowerOnTime	✓	✓
70	cpuPowerStretch	CPU Power Stretch	✓	X
71	motherBoardPowerOnTime	MotherBoard PowerOnTime	✓	✓

	Target	Title	/get	/set
72	emOnTime	EM On Time	✓	✓
73	emPressTrip	EM Pressure Trip	✓	✓
74	filaments (array indexed by 0, maxItems: 2)		✓	✓
75	emisOnTime	Filament Type	✓	✓
76	emisPressTRip	Filament Type	✓	✓
77	peakfind (array indexed by 1, maxItems: 32)		✓	✓
78	peakfindStatus	Peakfind Status	✓	✓
79	userDB	User DataBase	✓	✓
80	s5	S5	✓	✓
81	s6	S6	✓	✓
82	s7	S7	✓	✓
83	s8	S8	✓	✓
84	diagnosticData	Diagnostic Data	✓	✓
85	internalBoxTemperature	Internal Box Temperature	✓	X
86	anodePotential	Anode potential	✓	X
87	emissionCurrent	Emission current	✓	X
88	focusPotential	Focus potential	✓	X
89	electronEnergy	Electron Energy	✓	X
90	filamentPotential	Filament potential	✓	X
91	filamentCurrent	Filament current	✓	X
92	electronMultiplierPotential	Electron Multiplier potential	✓	X
93	measurement	Measured Data	✓	✓
94	totalPressure	Total Pressure	✓	X
95	scans (array indexed by 0)	Versatile Scan Data	✓	X
96	scannum		✓	✓
97	scansize		✓	✓
98	values (array indexed by 0, maxItems: 16384)		✓	✓
99	scansPow2 (array indexed by 0)	Versatile Scan Data - Pow2 Format	✓	X
100	scannum		✓	✓
101	scansize		✓	✓
102	values (array indexed by 0, maxItems: 16384)		✓	✓
103	binaryScans	Versatile Scan Data - Binary Format	✓	X
104	data	Versatile Data - JSON Format	✓	X
105	start		✓	✓
106	scansize		✓	✓
107	values (array indexed by 0, maxItems: 16384)		✓	✓
108	dataPow2	Versatile Data - Pow2 Format	✓	X
109	start		✓	✓

	Target	Title	/get	/set
110	scansize		✓	✓
111	values (array indexed by 0, maxItems: 16384)		✓	✓
112	binaryData	Versatile Data - Binary Format	✓	X
113	nextScan	Next Scan	✓	X
114	start		✓	✓
115	scansize		✓	✓
116	values (array indexed by 0, maxItems: 16384)		✓	✓
117	nextScanNumber	Next Scan Number	✓	✓
118	binaryNextScan	Binary Next Scan	✓	X
119	scanInfo	Scan Info	✓	✓
120	firstScan	First Scan	✓	X
121	lastScan	Last Scan	✓	X
122	currentScan	Current Scan	✓	X
123	pointsPerScan	Points Per Scan	✓	X
124	pointsInCurrentScan	Points In Current Scan	✓	X
125	scanning	Scanning	✓	X
126	sensorDetector	Sensor Detector parameters	✓	✓
127	emVoltageMax	EM Voltage Max	✓	X
128	emVoltageMin	EM Voltage Min	✓	X
129	emVoltage	EM Voltage	✓	✓
130	emGain	EM Gain	✓	✓
131	emGainMass	EM Gain Mass	✓	✓
132	defaultLoad	Load Default Parameters	X	✓
133	factoryLoad	Load Factory Parameters	X	✓
134	factorySave	Save Factory Parameters	X	✓
135	userLoad	Load User Parameters	X	✓
136	userSave	Save User Parameters	X	✓
137	sensorIonSource	Sensor Ion Source parameters	✓	✓
138	emissionCurrentMax	Max Emission Current	✓	X
139	emissionCurrentMin	Min Emission Current	✓	X
140	anodeMax	Anode Voltage Upper Bound	✓	X
141	anodeMin	Anode Voltage Lower Bound	✓	X
142	focusMax	Max Focus Voltage	✓	X
143	focusMin	Min Focus Voltage	✓	X
144	electronEnergyMax	Electron Energy Upper Bound	✓	X
145	electronEnergyMin	Electron Energy Lower Bound	✓	X
146	ionEnergyMax	Ion Energy Max	✓	X
147	ionEnergyMin	Ion Energy Min	✓	X

	Target	Title	/get	/set
148	structuresMax	Structures Max	✓	X
149	activate	Activate	X	✓
150	filamentSelected	Selected Filament	✓	✓
151	filamentSelectedOverride	Selected Filament Override	✓	✓
152	workingStructure	Working Structure	✓	✓
153	emissionLevel	Emission Level	✓	✓
154	optimizationType	Optimization Type	✓	✓
155	filamentPreferred	Filament Preferred	✓	✓
156	nomUserCopy	Nominal To User Copy	X	✓
157	nomFactoryCopy	Nominal to Factory Copy	X	✓
158	userTablesValid	User Tables Valid	✓	X
159	factoryTablesValid	Factory Tables Valid	✓	X
160	workingLoadSrc	Working Load Source	✓	X
161	emissionCurrent	Emission current	✓	✓
162	anode	Anode Voltage	✓	✓
163	focus	Focus Voltage	✓	✓
164	electronEnergy	Electron Energy	✓	✓
165	ionEnergyGlobal	Ion Energy Global	✓	✓
166	normalizationFactor	Normalization Factor	✓	✓
167	ppLinConst1	PP Linearization Constant 1	✓	✓
168	ppLinConst2	PP Linearization Constant 2	✓	✓
169	ppSensitivityFactor	PP Sensitivity Factor	✓	✓
170	tPThresholdCurrent	Total Pressure Emission Trip Threshold Current	✓	✓
171	tPTreshholdCount	Total Pressure Emission Trip Threshold Count	✓	✓
172	tPTripCount	Total Pressure Emission Trip Count	✓	✓
173	tPEMThresholdCurrent	Total Pressure EM Trip Threshold Current	✓	✓
174	tPEMThresholdCount	Total Pressure EM Trip Threshold Count	✓	✓
175	tPEMTripCount	TP EM Trip Count	✓	✓
176	tPCalPointsMax	Total Pressure Calibration Points Max	✓	X
177	tPCalPointsMin	Total Pressure Calibration Points Min	✓	X
178	tPCalPointsInUse	Total Pressure Calibration Points in Use	✓	✓
179	tPCalPressSet	Total Pressure Calibration Pressure Set	✓	✓
180	tPCalPressRemove	Total Pressure Calibration Pressure	✓	✓

	Target	Title	/get	/set
		Remove		
181	tPCalPointRemove	Total Pressure Calibration Point Remove	✓	✓
182	tPCalTableShift	Total Pressure Calibration Table Shift	✓	✓
183	tPunits	TP units	✓	✓
184	degasDuration	Degas Duration	✓	✓
185	emisRestartDelay	Emis Restart Delay	✓	✓
186	emisRestartCount	Emis Restart Count	✓	✓
187	emisRestartType	Emis Restart Type	✓	✓
188	calIndex (array indexed by 0, maxItems: 10)		✓	✓
189	tPCalPointPress	Total Pressure Calibration Point Pressure	✓	✓
190	tPCalPointCurrent	Total Pressure Calibration Point Current	✓	✓
191	tPCalPointSet	Total Pressure Calibration Point Set	✓	✓
192	ionSource (array indexed by 0)		✓	✓
193	defaultLoad	Load Default Parameters	X	✓
194	factoryLoad	Load Factory Parameters	X	✓
195	userLoad	Load User Parameters	X	✓
196	userSave	Save User Parameters	X	✓
197	sensorFilter	Sensor Filter Parameters	✓	✓
198	massMax	Tune Max Mass	✓	X
199	massMin	Tune Min Mass	✓	X
200	dwellMax	Max Allowable Dwell	✓	X
201	dwellMin	Min Allowable Dwell	✓	X
202	designMassMax	Design Max Mass	✓	X
203	designMassMin	Design Min Mass	✓	X
204	rodPolarity	Rod Polarity	✓	✓
205	workingTune	Working Tune	✓	✓
206	maxDwell	Max Allowable Dwell (fixed)	✓	X
207	minDwell	Min Allowable Dwell (fixed)	✓	X
208	scanSetup	Versatile Scan Parameters	✓	✓
209	maxChannels	Versatile Max Scan Channels	✓	X
210	startChannel	Scan Start Channel	✓	✓
211	stopChannel	Scan Stop Channel	✓	✓
212	scanCount	Scan Count	✓	✓
213	scanStart	Scan Start	X	✓
214	scanStop	Scan Stop	✓	✓

	Target	Title	/get	/set
215	scanInterval	Scan Interval	✓	✓
216	dwellGlobal	Dwell Global	✓	✓
217	leadInDwell	Lead In Dwell	✓	✓
218	leakCheckMass	Leak Check Mass	✓	✓
219	scanTimeTotals	Scan Time Total	✓	✓
220	channel (array indexed by 1)		✓	✓
221	channelType	Channel Type	✓	✓
222	startMassRaw	Raw Channel # Start Mass or flag	✓	✓
223	stopMassRaw	Raw Channel # Stop Mass or special	✓	✓
224	ppamu	Channel # points per AMU	✓	✓
225	dwell	Channel # Dwell	✓	✓
226	emVoltage	Channel # Local EM voltage	✓	✓
227	focusVoltage	Channel # Local Focus voltage	✗	✗
228	ionEnergy	Channel # Local Ion Energy	✗	✗
229	extra	Channel # Extra	✓	✓
230	leadIn	Channel # Lead In	✓	✓
231	enabled	Channel # Measurement Enabled	✓	✓
232	scaleFactor	Channel # Scale Factor	✓	✓
233	equivIonFactor	Channel # Equivalent Ion Current	✓	✓
234	digOutNum	Channel # Digital Output to Map	✓	✓
235	digOutThresUpper	Digital Output Upper Threshold	✓	✓
236	digOutThresLower	Digital Output Lower Threshold	✓	✓
237	startMass	Channel # Start Mass	✓	✓
238	stopMass	Channel # Stop Mass	✓	✓
239	channelMode	Channel Mode	✓	✓
240	aONum	Channel # Analog Output to Map	✓	✓
241	aOInputLowLimit	Analog Output Input Low Limit	✓	✓
242	aOInputHighLimit	Analog Output Input High Limit	✓	✓
243	aOMode	Analog Output Mode	✓	✓
244	analogInput	Analog Input Parameters	✓	✓
245	ioChannelCount	Analog Input Channels	✓	X
246	defaultLoad	Analog Input Default Load	X	✓
247	eventsDefaultLoad	Analog Input Events Default Load	X	✓
248	channelsAvailable	Analog Input Channels Available	✓	X
249	eventCount	Event Count	✓	X
250	aiEvent (array indexed by 0, maxItems: 32)		✓	✓
251	eventMapping	Event to AI mapping	✓	✓
252	eventThreshold1	Event Threshold 1	✓	✓

	Target	Title	/get	/set
253	eventMode1	Event Mode 1	✓	✓
254	eventAction1	Event Action 1	✓	✓
255	eventActionCount1	Event Action Count 1	✓	✓
256	eventThreshold2	Event Threshold 2	✓	✓
257	eventMode2	Event Mode 2	✓	✓
258	eventAction2	Event Action 2	✓	✓
259	eventActionCount2	Event Action Count 2	✓	✓
260	eventThreshold3	Event Threshold 3	✓	✓
261	eventMode3	Event Mode 3	✓	✓
262	eventAction3	Event Action 3	✓	✓
263	eventActionCount3	Event Action Count 3	✓	✓
264	eventThreshold4	Event Threshold 4	✓	✓
265	eventMode4	Event Mode 4	✓	✓
266	eventAction4	Event Action 4	✓	✓
267	eventActionCount4	Event Action Count 4	✓	✓
268	ioChannel (array indexed by 0, maxItems: 32)		✓	✓
269	rawValue	Analog Input y Raw value	✓	X
270	designOffset	Analog Input y Design Offset	✓	✓
271	errorOffset	Analog Input y Error Offset	✓	✓
272	designMultiplier	Analog Input y Design Multiplier	✓	✓
273	errorMultiplier	Analog Input y Error Multiplier	✓	✓
274	correctedValue	Analog Input y Corrected value	✓	X
275	mode	Analog Input y Mode	✓	✓
276	scaleMsgCount	Analog Input y Scale message count	✗	✗
277	scaleMsgId	Analog Input y Scale message Id(s)	✗	✗
278	scaledValue	Analog Input y Scaled value	✓	X
279	bypass	Analog Input y Bypass	✓	✓
280	appValue	Analog Input y Application Value	✓	✓
281	analogOutput	Analog Output Parameters	✓	✓
282	ioChannelCount	Analog Output Channels	✓	X
283	defaultLoad	Analog Outputs Default Load	X	✓
284	channelsAvailable	Analog Output Channels Available	✓	X
285	ioChannel (array indexed by 0, maxItems: 32)		✓	✓
286	raw	Analog Output # Raw	✓	✓
287	bypass	Analog Output # Bypass	✓	✓
288	scaleInValue	Analog Output # Scale In Value	✓	✓
289	scaleMode	Analog Output # Scale Mode	✓	✓
290	ceiling	Analog Output # Ceiling {Max}	✓	✓

	Target	Title	/get	/set
291	floor	Analog Output # Floor {Min}	✓	✓
292	scaleOutValue	Analog Out y Scale Out Value	✓	✓
293	designOffset	Analog Output # Design Offset	✓	✓
294	errorOffset	Analog Output # Error Offset	✓	✓
295	designGain	Analog Output # Design Gain	✓	✓
296	errorGain	Analog Output # Error Gain	✓	✓
297	hardwareValue	Analog Output # Hardware Value	✓	✓
298	inputLowLimit	Analog Output # Input Low Limit	✓	✓
299	inputHighLimit	Analog Output # Input High Limit	✓	✓
300	digitalInput	Digital Input Parameters	✓	✓
301	ioChannelCount	Digital Input Channels	✓	X
302	bitConfig	Extended IO Bit Config	✓	✓
303	rawValueBitwise	Raw Value Bitwise	✓	X
304	defaultLoad	Digital Input Default Load	X	✓
305	channelsAvailable	Digital Input Channels Available	✓	X
306	hardwareValues	Digital Input Hardware Values	✓	X
307	ioChannel (array indexed by 0, maxItems: 32)		✓	✓
308	rawValue	Digital Input # Raw value	✓	X
309	trigger1Mode	Trigger 1 Mode	✓	✓
310	trigger1Action	Trigger 1 Action	✓	✓
311	trigger1ActCnt	Trigger 1 Action Count	✓	✓
312	trigger2Mode	Trigger 2 Mode	✓	✓
313	trigger2Action	Trigger 2 Action	✓	✓
314	trigger2ActCnt	Trigger 2 Action Count	✓	✓
315	trigger3Mode	Trigger 3 Mode	✓	✓
316	trigger3Action	Trigger 3 Action	✓	✓
317	trigger3ActCnt	Trigger 3 Action Count	✓	✓
318	trigger4Mode	Trigger 4 Mode	✓	✓
319	trigger4Action	Trigger 4 Action	✓	✓
320	trigger4ActCnt	Trigger 4 Action Count	✓	✓
321	digitalOutput	Digital Output Parameters	✓	✓
322	ioChannelCount	Digital Output Channels	✓	X
323	bitConfig	Extended IO Bit Config	✓	✓
324	defaultLoad	Digital Output Default Load	X	✓
325	channelsAvailable	Digital Output Channels Available	✓	X
326	manualValues	Digital Output Manual Values	✓	✓
327	valveControlValues	Digital Output Valve Control Values	✓	✓
328	logicValues	Digital Output Logic Values	✓	✓

	Target	Title	/get	/set
329	hardwareValues	Digital Output Hardware Values	✓	✓
330	nativeRelayCycles	Digital Output Native Relay Cycles	✓	✓
331	ioChannel (array indexed by 0, maxItems: 32)		✓	✓
332	mode	Digital Output # Mode	✓	✓
333	invert	Digital Output # Invert	✓	✓
334	logicValue	Digital Output # Logic Value	✓	✓
335	hardwareValue	Digital Output # Hardware value	✓	✓
336	ssAllNotMask	Digital Output # System Status All Not Mask	✓	✓
337	ssAllMask	Digital Output # System Status All Mask	✓	✓
338	ssAnyNotMask	Digital Output # System Status Any Not Mask	✓	✓
339	ssAnyMask	Digital Output # System Status Any Mask	✓	✓
340	internals	Internal Variables	✓	✓
341	bootCount	Boot Count	✓	✓
342	diagnostics	Diagnostics	✓	✓
343	anomalyStateSaveTriggerConditions	Anomaly state save trigger conditions	✗	✗
344	gauge	Gauge	✓	✓
345	gaugeControl	Gauge Control	✓	✓
346	gaugeState	Gauge State	✓	X
347	gaugePressure	Gauge Pressure	✓	X
348	gaugeOperMode	Gauge Operational Mode	✓	✓
349	pressureUnits	Pressure Units	✓	✓
350	gaugeName	Gauge Name	✓	✓
351	degas	External Gauge Degas Control	✓	✓

Legend

X Operation not possible.

✓ Operation implemented.

✗ Operation not yet implemented on box, but should work in emulator.

Note: To “set” API’s is at your own risk!

XXX [Use Links:](#)

Example:
How to following the Links

0. API root

[/](#)

This server provides a hierarchy of targets that can be read or modified over HTTP. To read target values, clients send HTTP requests to the target name appended with **/get**. To modify target subproperties, clients append the target name with **/set?** followed by key-value pairs for the subproperties to modify. For example a request to **/point/set?x=42&y=100** sets the values of targets **/point/x** and **/point/y**, while **/point/z**, if it exists, remains unchanged.

Targets flagged as **writeonly** can be **/set** but not **/get**. Targets flagged as **exec** are executed by setting them to any value (e.g. 1, or run).

Responses are returned in the HTTP body as JSON objects with properties **name**, **data** and **origin**. To view responses in Firefox or Chrome, it is useful to install JSONView. To test this API from a command line use curl. A machine-readable JSON Schema description of this API is available from **/get.api**

Property names below that begin with \$ (in the right hand column) are used by the server, and can safely be ignored by the client.

type: object

1. Modular Mass Spec

[/mmsp](#)

Each target has a \$block (0x00-0xFF) and \$message number (0x00-0xFF) e.g. **/mmsp/electronicsInfo/serialNumber** has \$message 0x0010.

type: object

version: 1.5.2

2. Electronics Information

</mmssp/electronicsInfo>

type: object \$block: 0x01XX

3. Serial Number

</mmssp/electronicsInfo/serialNumber>

This target returns a 8 digit numeric string containing the Serial Number found on the MMSP label.
Example "71234567".

type:	string	\$message:	0x10
version:	0.5.0	\$type:	STR
minLength:	0	\$emul:	C7TS41D05371
maxLength:	80	\$flags:	PUBLIC
		\$length:	0:80
		\$readmode:	normal
		\$writemode:	admin
		\$execmode:	null

4. Monitor Firmware Version

</mmssp/electronicsInfo/monitorFWVersion>

This target returns a variable length string that contains the monitor's firmware version.
"X.YY.ZZ"

Where:

'X' is an integer indicating the major level

'YY' is an integer indicating the minor level

'ZZ' is an integer which indicates the beta level if not '0'

Example: "1.00.00" (the first released version).

type:	string	\$message:	0x14
version:	0.5.0	\$type:	STR
readonly:	true	\$emul:	1.0.0
minLength:	0	\$flags:	PUBLIC
maxLength:	80	\$length:	0:80
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

5. Control Firmware Version

[/mmsp/electronicsInfo/controlFWVersion](#)

This target returns a variable length string that contains the control processor's firmware version.
"X.YY.ZZ"

Where:

'X' is an integer indicating the major level

'YY' is an integer indicating the minor level

'ZZ' is an integer which indicates the beta level if not '0'

Example: "1.00.00" (the first released version).

type:	string	\$message:	0x15
version:	0.5.0	\$type:	STR
readonly:	true	\$emul:	1.0.0
minLength:	0	\$flags:	PUBLIC
maxLength:	40	\$length:	0:40
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

6. Genus

[/mmsp/electronicsInfo/genus](#)

Product Genus. MPH=1, MPE=2, MPS=3, MPP=4, MPCPM=5, MPHPR=6, XPR=7, MPPROSPECTOR=8, MPCPMSPS=9, MPCPMADV=10, SPS_ADV=11, XPR_ADV=12

type:	integer	\$message:	0x37
version:	0.5.0	\$type:	U8
minimum:	1	\$emul:	0
maximum:	6	\$flags:	PUBLIC
		\$range:	1:6
		\$readmode:	normal
		\$writemode:	admin
		\$execmode:	null

7. Mass Range

[/mmsp/electronicsInfo/massRange](#)

Range in AMU e.g. 100, 200, 300

type:	integer	\$message:	0x44
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	200
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

8. Communications Port Configuration

[/mmsp/communication](#)

The MMSP supports simultaneous connection of more than one serial interface. In order to do this a control access scheme was devised. The scheme allows only one port based on priority to have control of the MMSP. Three states are defined for each communications port. The states are:

Absent

- No hardware connection to port or no property formulated messages for more than one time-out period.

Monitor

- At least one properly formulated message within a time-out period. When in this state the port may be used to watch the system, but no parameter updates or control functions are allowed. The sole exception is to set the port's control request. The COR (Clear On Read) bit in the MCB (Message Control Byte) will be ignored.

Control

- At least one properly formulated message within a time-out period and the highest priority port requesting control. The port priority is set when the control access is requested. When in this state full access to the MMSP is allowed.

Figure

2

Communication Port States

type: object \$block: 0x02XX

9. IP Address

</mmsp/communication/ipAddress>

IP Address stored in 1-wire

type:	string	\$message:	0x05
version:	0.5.0	\$type:	STR
minLength:	0	\$emul:	192.168.1.1
maxLength:	32	\$flags:	PUBLIC
		\$length:	0:32
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

10. MMSP MAC Address

</mmsp/communication/macAddress>

MAC Address stored in 1-wire

type:	string	\$message:	0x06
version:	0.5.0	\$type:	STR
minLength:	0	\$emul:	AB:CD:EF:01:23:45
maxLength:	64	\$flags:	PUBLIC
		\$length:	0:64
		\$readmode:	normal
		\$writemode:	admin
		\$execmode:	null

11. MMSP Gateway

</mmsp/communication/gateway>

Gateway stored in 1-wire

type:	string	\$message:	0x07
version:	0.9.0	\$type:	STR
minLength:	0	\$flags:	PUBLIC
maxLength:	32	\$length:	0:32
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

12. MMSP Mask

</mmsp/communication/mask>

Subnet Mask stored in 1-wire

type:	string	\$message:	0x08
version:	0.9.0	\$type:	STR
minLength:	0	\$flags:	PUBLIC
maxLength:	32	\$length:	0:32
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

13. MMSP Port

</mmsp/communication/port>

Port stored in 1-wire

type:	integer	\$message:	0x09
version:	0.17.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

14. Boot from DHCP

[/mmssp/communication/dhcp](#)

Sets 1-wire byte owRGA_CPU_TCPIP.DHCP to either enable or disable the MMSP from obtaining its IP address using DHCP.

type:	integer	\$message:	0x0D
version:	0.47.0	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

15. Session ID

[/mmssp/communication/sessionID](#)

Returns the Communications current Session ID.

type:	string	\$message:	0x20
version:	0.22.0	\$type:	STR
		\$flags:	SPECIAL PUBLIC
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

16. Issue Log

[/mmssp/communication/issueLog](#)

Returns and resets the complete current error log.

type:	object	\$message:	0x21
version:	0.45.0	\$type:	STRUCT
atomic:	true	\$flags:	SPECIAL PUBLIC
		\$readmode:	locked
		\$writemode:	service
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target.

17. Error Log

[/mmssp/communication/errorLog](#)

Returns the current error log.

type:	object	\$message:	0x22
version:	0.33.0	\$type:	STRUCT
readonly:	true	\$flags:	SPECIAL PUBLIC
atomic:	true	\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target.

18. Session Timeout

[/mmssp/communication/sessionTimeout](#)

Amount of time a session can have no communication before timing out. Default is 60 sec; 0 means "do not time out".

type:	number	\$baseunits:	msec
version:	0.47.0	\$message:	0x24
divisibleBy:	0.001	\$type:	U32
units:	sec	\$flags:	SPECIAL PUBLIC
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

19. Identify MMSP Location using the CPU Power LED

[/mmssp/communication/locationIdentifyCPU](#)

Writing a 1 to this location will cause the CPU board Power LED to start blinking. This allows the blinking CPU board to be identified. Writing a 0 will turn the LED off.

type:	integer	\$message:	0x31
version:	0.9.0	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

20. Login

</mmsp/communication/login>

Provides Login access/information.

type:	string	\$message:	0x63
version:	0.17.0	\$type:	STR
		\$flags:	SPECIAL PUBLIC
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

21. Client Name

</mmsp/communication/clientName>

Name of current client (e.g. "PVMassSpec", "LabView",...)

type:	string	\$message:	0x70
version:	0.14.0	\$type:	STR
minLength:	0	\$flags:	SPECIAL PUBLIC
maxLength:	20	\$length:	0:20
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

22. Control

</mmsp/communication/control>

Take or release control.

You may use one of the special verbs "request", "take", "release" or "force". Or you may set control to one of them:

```
/mmsp/communication/control/take
/mmsp/communication/control/set?take
```

- Issuing a "request" takes control of the instrument provided no other client currently has control. The "request" will return an error if another session is in control.
- Issuing a "take" takes control of the instrument provided no other client currently has control or the control client has lower privilege. The "take" will return an error if another session of equal or greater privilege is in control.

- Issuing a "force" forcibly takes control even if another client has control. Clients cannot force control from a higher privileged client.
- Issuing a "release" releases control.

Issuing any command that requires control performs an implicit "request". The command will fail if control cannot be taken.

Clients should "release" control whenever they enter an inactive period.

	No one in control	Self in control	Lower privilege in control	Equal privilege in control	Higher privilege in control
Set command	Succeed	Succeed	Fail	Fail	Fail
REQUEST	Succeed	Succeed	Fail	Fail	Fail
TAKE	Succeed	Succeed	Succeed	Fail	Fail
FORCE	Succeed	Succeed	Succeed	Succeed	Fail
RELEASE	Succeed	Succeed	Fail	Fail	Fail

type:	string	\$message:	0x80
version:	0.14.0	\$type:	STR
minLength:	0	\$flags:	SPECIAL PUBLIC
maxLength:	16	\$length:	0:16
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

23. Control Info

</mmsp/communication/controlInfo>

Returns information about the current controlling session, or null if no session is in control:

- sessionID - session id of the controller
- ipAddress - IP Address of the client (may not be accurate if going through a proxy).
- canForce - true if a force will succeed
- secondsSinceLastRequest - time since last request
- secondsSinceLastControlRequest - time since last control request
- secondsSinceLastDataRequest - time since last data request
- clientName - name of client, or empty string if client is not providing a name

type:	object	\$message:	0x81
version:	0.14.0	\$type:	STRUCT
readonly:	true	\$flags:	SPECIAL PUBLIC
atomic:	true	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target.

24. sessionID

</mmsp/communication/controlInfo/sessionID>

type: string
readonly: true

25. ipAddress

</mmsp/communication/controlInfo/ipAddress>

type: string
readonly: true

26. amInControl

</mmsp/communication/controlInfo/amInControl>

type: boolean
readonly: true

27. canTakeControl

</mmsp/communication/controlInfo/canTakeControl>

type: boolean
readonly: true

28. canForceControl

</mmsp/communication/controlInfo/canForceControl>

type: boolean

readonly: true

29. secondsSinceLastRequest

</mmsp/communication/controlInfo/secondsSinceLastRequest>

type: integer

readonly: true

30. secondsSinceLastControlRequest

</mmsp/communication/controlInfo/secondsSinceLastControlRequest>

type: integer

readonly: true

31. secondsSinceLastDataRequest

</mmsp/communication/controlInfo/secondsSinceLastDataRequest>

type: integer

readonly: true

32. clientName

</mmsp/communication/controlInfo/clientName>

type: string

readonly: true

33. Am In Control

</mmsp/communication/amInControl>

Returns true if current session is in control.

type:	boolean	\$message:	0x82
version:	0.14.0	\$type:	BOOL
readonly:	true	\$flags:	SPECIAL PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

34. ControlLock

</mmsp/communication/controlLock>

Returns true if current session is in control.

type:	string	\$message:	0x90
version:	0.14.0	\$type:	U8
enum:	Unlocked,Session,System,Locked	\$flags:	SPECIAL PUBLIC
		\$range:	Unlocked,Session,System,Locked
		\$readmode:	normal
		\$writemode:	protected
		\$execmode:	null

35. General control

[/mmssp/generalControl](#)

These targets perform general instrument control

type: object \$block: 0x04XX

36. Set Emission State

[/mmssp/generalControl/setEmission](#)

This target requests the emission and RF states be set to On or Off. First the emission state is requested. If the emission request is accepted it then requests the RF state.

This target is equivalent to sending:

For On:

EmissionSet On

RFGenSweep

For Off:

EmissionSet Off

RFGeneratorSet Off

type:	string	\$message:	0x10
version:	0.5.0	\$type:	U8
enum:	Off,On	\$flags:	PUBLIC
		\$range:	Off,On
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

37. Set Degas State

[/mmssp/generalControl/degas](#)

This target sets the Degas state to On or Off. Once started, the degas runs with specified sensor parameters for two minutes (default time).

type:	string	\$message:	0x12
version:	0.5.0	\$type:	U8
enum:		\$flags:	PUBLIC
		\$range:	
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

38. Set Electron Multiplier State

[/mmsp/generalControl/setEM](#)

This target requests the Electron Multiplier state of On or Off. The Electron Multiplier does not start immediately, so it's important when sending an On request to verify the state by reading the "EM on" bit in the System Status message. If the Electron Multiplier is requested but has not yet started and has not failed then the "EM requested" bit will be set.

Once the Electron Multiplier is on and in control the "EM on" bit will be set. If the Electron Multiplier fails to start the "EM failed" bit will be set in the Hardware Errors message.

An Electron Multiplier Off request can be verified by the "EM on" bit in System Status message being clear.

type:	string	\$message:	0x20
version:	0.5.0	\$type:	U8
enum:	Off,On	\$flags:	PUBLIC
		\$range:	Off,On
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

39. RF Generator Set

[/mmsp/generalControl/rfGeneratorSet](#)

This target enables or disables the Quadrupole Mass Filter RF and DC generator.

type:	integer	\$message:	0x30
version:	0.5.1	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

40. Baseline Measure Set

[/mmsp/generalControl/baselineMeasureSet](#)

This target enables or disables the scheduling of baseline measurements during a scan. These measurements are used to correct for errors related to temperature, humidity and leakage. These measurements are not included in returned scan data.

type:	integer	\$message:	0x40
version:	0.5.1	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

41. Baseline Correction Set

[/mmssp/generalControl/baselineCorrectSet](#)

This target enables or disables the correction of raw data with the most recent data collect by Baseline Measure.

type:	integer	\$message:	0x41
version:	0.5.1	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

42. Peakfind Measure State Set

[/mmssp/generalControl/peakfindMeasureSet](#)

This target enables or disables the scheduling of peakfind measurements during a scan. These measurements can be used, if PeakfindCorrect is enabled, to correct for errors related to peak position change. These measurements are not included in returned scan data.

type:	integer	\$message:	0x42
version:	0.5.1	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

43. Peakfind Correct Set

[/mmssp/generalControl/peakfindCorrectSet](#)

This target enables or disables the correction of measured peak location in single point per amu channels. The most recent data collected by PeakfindMeasure is used.

type:	integer	\$message:	0x43
version:	0.5.1	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

44. EM Equivalent Ion Current Set

</mmsp/generalControl/emEquivIonSet>

This target enables or disables the conversion of electron current from the electron multiplier detector to equivalent ion current.

type:	integer	\$message:	0x44
version:	0.8.0	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

45. Partial Pressure Linearization

</mmsp/generalControl/ppLinearization>

This target enables or disables Partial Pressure Linearization based on Total Pressure.

type:	integer	\$message:	0x45
version:	0.9.0	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

46. Fan State

</mmsp/generalControl/fanState>

This target gets or sets the state of the electronics box cooling fan disabled.

0 = Fan is on, not disabled

1 = Fan is off, disabled

type:	integer	\$message:	0x48
version:	1.0.10	\$type:	U8
minimum:	0	\$flags:	PUBLIC
maximum:	1	\$range:	0:1
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

47. User Timer 1

[/mmsp/generalControl/userTimer1](#)

This target Gets or Sets the 32-bit User Timer 1. This timer increments at a 1ms rate.

Notes:

This timer will rollover every 49.7 days if not reset in less time.

This timer starts from zero on electronics power up.

type:	integer	\$message:	0x71
version:	0.7.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

48. User Timer 2

[/mmsp/generalControl/userTimer2](#)

This target Gets or Sets the 32-bit User Timer 2. This timer increments at a 1ms rate.

Notes:

This timer will rollover every 49.7 days if not reset in less time.

This timer starts from zero on electronics power up.

type:	integer	\$message:	0x73
version:	0.7.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

49. Shutdown

[/mmsp/generalControl/shutdown](#)

This target shuts down the Emission, RF and (if part of a system) Heaters and inlet valves.

type:	integer	\$message:	0x80
version:	0.5.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

50. Sensor Information

[/mmssp/sensorInfo](#)

type: object \$block: 0x05XX

51. Sensor Name

[/mmssp/sensorInfo/name](#)

The Sensor ID, stored in the SD card.

type:	string	\$message:	0x10
version:	0.5.0	\$type:	STR
minLength:	0	\$emul:	MMSP Emulator 1
maxLength:	20	\$flags:	PUBLIC
		\$length:	0:20
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

52. Sensor Description

[/mmssp/sensorInfo/description](#)

The Sensor Description, stored in the SD card.

type:	string	\$message:	0x11
version:	0.5.0	\$type:	STR
minLength:	0	\$emul:	MMSP Emulator
maxLength:	40	\$flags:	PUBLIC
		\$length:	0:40
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

53. Sensor Serial Number

[/mmsp/sensorInfo/serialNumber](#)

Sensor serial number

type:	integer	\$message:	0x39
version:	0.5.0	\$type:	U32
		\$emul:	1234
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

54. Ion Source Type

[/mmsp/sensorInfo/ionSourceType](#)

Enumerated Ion Source Type: 0=open, 1=closed, 2=grid, 3=crossbeam

type:	integer	\$message:	0x41
version:	0.5.0	\$type:	U8
minimum:	0	\$flags:	PUBLIC
maximum:	3	\$range:	0:3
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

55. Mass Filter Type

[/mmsp/sensorInfo/massFilterType](#)

Enumerated Mass filter type: 0=Standard

type:	integer	\$message:	0x43
version:	0.5.0	\$type:	U8
minimum:	0	\$flags:	PUBLIC
maximum:	1	\$range:	0:1
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

56. Detector Type

[/mmsp/sensorInfo/detectorType](#)

Enumerated detector type: 0=FC, 1=CDEM, 2=MCP, 3=366B

type:	string	\$message:	0x45
version:	0.5.0	\$type:	U8
enum:		\$emul:	1
		\$flags:	PUBLIC
		\$range:	
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

57. filaments

[/mmsp/sensorInfo/filaments](#)

type: array \$range: 0:2

minIndex: 0

minItems: 0

maxItems: 2

Array of objects with following properties:

type: object

58. Filament Type

[/mmsp/sensorInfo/filaments/<index#>/filamentType](#)

Enumerated Filament Type: 0=Tungsten, 1=Yttria coated Iridium

type:	integer	\$message:	0x42
version:	0.5.0	\$type:	U8
minimum:	0	\$index:	Filaments
maximum:	1	\$flags:	PUBLIC
		\$range:	0:1
		\$length:	1:2
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

59. Status

[/mmssp/status](#)

These targets return the MMSP status

type: object \$block: 0x06XX

60. System Status

[/mmssp/status/systemStatus](#)

The System Status gives a single message status of the primary MMSP functions.

The bit definitions are tabulated below.

Bit(s)	Description
31	0/1 = Emission regulator, regulated current mode Off/On
30	0/1 = Emission regulator, constant power mode Off/On
29	0/1 = Emission regulator, degas mode Off/On
28	0/1 = Emission regulator, request pending No/Yes
27	0/1 = Emission interlock active No/Yes
26	0/1 = Selected filament is 1/2
25	0/1 = RF generator Off/On
24	0/1 = Rod polarity Normal/Reversed
23	0/1 = Electron multiplier Off/On
22	0/1 = Electron multiplier request pending No/Yes
21	0/1 = Electron multiplier interlock active No/Yes
20	0/1 = Global ion energy in use No/Yes
19:17	Reserved
16	0/1 = Status Relay Off/On
15	Reserved
14	0/1 = Total pressure based partial pressure linearization Off/On
13	0/1 = Peak position measurement scheduled No/Yes
12	0/1 = Peak position correction Off/On
11	0/1 = Baseline continuous measurement scheduled No/Yes
10	0/1 = Baseline single measurement scheduled Off/On
9	0/1 = Baseline correction Off/On
8	0/1 = RF generator high power re-resonate Off/On
7	0/1 = RF frequency change compensation Off/On
6	0/1 = RF board gain and offset compensation Off/On
5	0/1 = EM equivalent ion current Off/On
4	0/1 = Ion source normalization Off/On
3	0/1 = RF board temperature compensate setpointsOff/On

2	0/1 = RF generator always re-resonate Off/On
1	0/1 = Versiscan Active No/Yes
0	0/1 = Reset occurred since last query No/Yes

type:	integer	\$message:	0x10
version:	0.5.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

61. System Status 2

[/mmssp/status/systemStatus2](#)

The System Status 2 is a continuation of System Status.
The bit definitions are tabulated below.

Bit(s)	Description
31:2	Reserved
1	0/1 = Filament 2 Open No/Yes
0	0/1 = Filament 1 Open No/Yes

type:	integer	\$message:	0x11
version:	1.5.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

62. Hardware errors

[/mmssp/status/hardwareErrors](#)

The hardware errors message includes the primary PrismaPro MPP errors. Any error bit indicates a problem has occurred which may result in loss of data or hardware function.
The bit definitions are tabulated below. Writing a bit to one clears it. Writing a bit to zero

does not affect the bit's value. More than one bit can be cleared at a time. All the bits may be cleared by writing 0xFFFF. Recurring error conditions will reassert bits that are cleared.

Bit(s)	Description
31	0/1 = Emission No/Yes
30	0/1 = Electron multiplier No/Yes
29	0/1 = Anode No/Yes
28	0/1 = RF board No/Yes
27	0/1 = Total pressure No/Yes
26	0/1 = Internal power supply No/Yes
25	0/1 = Electronics box temperature No/Yes
24	0/1 = Electrometer No/Yes
23	0/1 = Database (any data block with integrity check) No/Yes
22	0/1 = Mass Filter No/Yes
21	0/1 = Ion Source No/Yes
20	0/1 = Detector No/Yes
19	0/1 = DDS No/Yes
18	0/1 = DSP communications No/Yes
17	0/1 = DEC communications No/Yes
16	0/1 = Peakfind No/Yes
15	0/1 = Filament Potential No/Yes
14	0/1 = Filament Current No/Yes
13	0/1 = Focus Potential No/Yes
12	0/1 = Cathode/EE Potential No/Yes
11	0/1 = DEC Control Excursion No/Yes
10:0	Reserved

type:	integer	\$message:	0x20
version:	0.5.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	locked
		\$execmode:	null

63. Filament 1 Open

[/mmssp/status/fil1Open](#)

Get or Set the state of the filament 1 open flag.

A Get value of zero indicates fillament is good, a value of 1 indactes filament is open.

Set to 0 to clear flag, set to 1 to set flag.

type:	integer	\$message:	0x23
version:	1.5.0	\$type:	U16
		\$flags:	NVM PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

64. Filament 2 Open

[/mmssp/status/fil2Open](#)

Get or Set the state of the filament 2 open flag.

A Get value of zero indicates fillament is good, a value of 1 indactes filament is open.

Set to 0 to clear flag, set to 1 to set flag.

type:	integer	\$message:	0x24
version:	1.5.0	\$type:	U16
		\$flags:	NVM PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

65. Hardware warnings

[/mmssp/status/hardwareWarnings](#)

The hardware warnings message includes the primary PrismaPro MPP hardware warnings.

Any warning bit indicates a potential problem may occur. The bit definitions are tabulated below. Writing a bit to one clears it. Writing a bit to zero does not affect the bit's value.

More than one bit can be cleared at a time. All the bits may be cleared by writing 0xFFFF.

Recurring warning conditions will reassert bits that are cleared.

Bit(s)	Description
31	0/1 = Emission No/Yes
30	0/1 = Electron multiplier No/Yes
29	0/1 = Anode No/Yes
28	0/1 = RF board No/Yes
27	0/1 = Total pressure No/Yes
26	0/1 = Internal power supply No/Yes
25	0/1 = Electronics box temperature No/Yes

24	0/1 = Electrometer No/Yes
23	0/1 = Database (any data block with integrity check) No/Yes
22	0/1 = Mass Filter No/Yes
21	0/1 = Ion Source No/Yes
20	0/1 = Detector No/Yes
19	0/1 = DDS No/Yes
18	0/1 = DSP communications No/Yes
17	0/1 = DEC communications No/Yes
16	0/1 = Peakfind No/Yes
15	0/1 = Filament Potential No/Yes
14	0/1 = Filament Current No/Yes
13	0/1 = Focus Potential No/Yes
12	0/1 = Cathode/EE Potential No/Yes
11	0/1 = DEC Control Excursion No/Yes
10:0	Reserved

type:	integer	\$message:	0x30
version:	0.5.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	locked
		\$execmode:	null

66. Power Supply PowerOnTime

</mmsp/status/powerSupplyPowerOnTime>

Get or Set the Power Supply board cumulative 'power on time' in seconds.

type:	integer	\$message:	0x70
version:	0.5.0	\$type:	U32
		\$emul:	4211
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	protected
		\$execmode:	null

67. EmissionStretch

</mmsp/status/emissionStretch>

Emission on time for the current stretch (i.e. the current duration since it was turned On) in seconds.

type:	integer	\$message:	0x71
version:	0.5.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

68. EMStretch

</mmsp/status/emStretch>

EM on time for the current stretch (i.e. the current duration since it was turned On) in seconds.

type:	integer	\$message:	0x72
version:	0.5.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

69. CPU PowerOnTime

</mmsp/status/cpuPowerOnTime>

Get or Set the CPU board cumulative 'power on time' in seconds.

type:	integer	\$message:	0x80
version:	0.5.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	protected
		\$execmode:	null

70. CPU Power Stretch

</mmssp/status/cpuPowerStretch>

Get CPU board 'power on time' since last reset in seconds.

type:	integer	\$message:	0x81
version:	0.62.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

71. MotherBoard PowerOnTime

</mmssp/status/motherBoardPowerOnTime>

Get or Set the motherboard cumulative 'power on time' in seconds.

type:	integer	\$message:	0xB0
version:	0.5.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	protected
		\$execmode:	null

72. EM On Time

</mmssp/status/emOnTime>

Get or Set the cumulative 'EM on time' in seconds.

type:	integer	\$message:	0xB3
version:	0.5.0	\$type:	U32
		\$emul:	1022
		\$flags:	PUBLIC
		\$length:	1:2
		\$readmode:	normal
		\$writemode:	protected
		\$execmode:	null

73. EM Pressure Trip

</mmssp/status/emPressTrip>

Number of EM over pressure trips for filaments A&B.

type:	integer	\$message:	0xB4
version:	0.5.0	\$type:	U16
		\$flags:	PUBLIC
		\$length:	1:2
		\$readmode:	normal
		\$writemode:	protected
		\$execmode:	null

74. filaments

</mmssp/status/filaments>

type:	array	\$range:	0:2
minIndex:	0		
minItems:	0		
maxItems:	2		

Array of objects with following properties:

type: object

75. Emission On Time

</mmssp/status/filaments/<index#>/emisOnTime>

Get or Set the cumulative 'emission on time' in seconds for filaments A&B.

type:	integer	\$message:	0xB1
version:	0.5.0	\$type:	U32
		\$emul:	1345
		\$index:	Filaments
		\$flags:	PUBLIC
		\$length:	1:2
		\$readmode:	normal
		\$writemode:	protected
		\$execmode:	null

76. Emission Pressure Trip

</mmsp/status/filaments/<index#>/emisPressTrip>

Number of emission over pressure trips for filaments A&B.

type:	integer	\$message:	0xB2
version:	0.5.0	\$type:	U16
		\$index:	Filaments
		\$flags:	PUBLIC
		\$length:	1:2
		\$readmode:	normal
		\$writemode:	protected
		\$execmode:	null

77. peakfind

</mmsp/status/peakfind>

type:	array	\$range:	1:650
minIndex:	1		
minItems:	0		
maxItems:	32		

Array of objects with following properties:

(Note this array begins with index **1**, so indexing by 0 is invalid)

type: object

78. Peakfind Status

</mmsp/status/peakfind/<index#>/peakfindStatus>

Peakfind status for a given mass in AMU. Indicates one or more of the following:

32:08	0/1 = Reserved
07	0/1 = Low: Peak location > 40 centiAMU below nominal No/Yes
06	0/1 = High: Peak location > 40 centiAMU above nominal No/Yes
05	0/1 = Wide: Max/Min amplitude less Peak Width Factor No/Yes
04	0/1 = None: No qualified peak found No/Yes
03	0/1 = Small: Max-Min amplitude less than Peak Size Factor No/Yes
02:01	0/1 = Reserved
00	0/1 = Good: A qualified usable peak location has been found. No/Yes

type:	integer	\$message:	0x61
version:	0.7.0	\$type:	U16
		\$index:	peakfind
		\$flags:	PUBLIC
		\$length:	1:650
		\$readmode:	normal
		\$writemode:	locked
		\$execmode:	null

79. User DataBase

[/mmsp/userDB](#)

User Database access

type: object \$block: 0x07XX

80. S5

[/mmsp/userDB/s5](#)

Write a string of up to 1024 characters to the SDRam card. File Location is the top level of the card.
File Name is SDFileDB.txt

type:	string	\$message:	0x15
version:	0.5.0	\$type:	STR
		\$flags:	PUBLIC
		\$length:	1024
		\$readmode:	control
		\$writemode:	control
		\$execmode:	null

81. S6

[/mmsp/userDB/s6](#)

Write a string of up to 1024 characters to the SDRam card. File Location is the top level of the card.
File Name is SDFileDB.txt

type:	string	\$message:	0x16
version:	0.5.0	\$type:	STR
		\$flags:	PUBLIC
		\$length:	1024
		\$readmode:	control
		\$writemode:	control
		\$execmode:	null

82. S7
[/mmssp/userDB/s7](#)

Write a string of up to 1024 characters to the SDRam card. File Location is the top level of the card.
File Name is SDFileDB.txt

type:	string	\$message:	0x17
version:	0.5.0	\$type:	STR
		\$flags:	PUBLIC
		\$length:	1024
		\$readmode:	control
		\$writemode:	control
		\$execmode:	null

83. S8
[/mmssp/userDB/s8](#)

Write a string of up to 1024 characters to the SDRam card. File Location is the top level of the card.
File Name is SDFileDB.txt

type:	string	\$message:	0x18
version:	0.5.0	\$type:	STR
		\$flags:	PUBLIC
		\$length:	1024
		\$readmode:	control
		\$writemode:	control
		\$execmode:	null

84. Diagnostic Data

[/mmssp/diagnosticData](#)

These targets return data internal to the MMSP. This data is primarily used for diagnostics
type: object \$block: 0x08XX

85. Internal Box Temperature

[/mmssp/diagnosticData/internalBoxTemperature](#)

Get internal box temperature in degrees C as measured on CPU board.

type:	number	\$message:	0x18
version:	0.6.0	\$type:	F32
readonly:	true	\$flags:	PUBLIC
units:	C	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

86. Anode potential

[/mmssp/diagnosticData/anodePotential](#)

Sensor Ion Source Anode Potential as measured on Power Supply board.

type:	integer	\$message:	0x30
version:	0.6.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
units:	cV	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

87. Emission current

[/mmssp/diagnosticData/emissionCurrent](#)

Sensor Ion Source Emission Current as measured on Power Supply board.

type:	integer	\$message:	0x31
version:	0.8.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
units:	uA	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

88. Focus potential

</mmsp/diagnosticData/focusPotential>

Sensor Ion Source Focus Potential as measured on Power Supply board.

type:	integer	\$message:	0x32
version:	0.6.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
units:	cV	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

89. Electron Energy

</mmsp/diagnosticData/electronEnergy>

Sensor Ion Source Electron Energy as measured on Power Supply board.

type:	integer	\$message:	0x33
version:	0.6.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
units:	cV	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

90. Filament potential

</mmsp/diagnosticData/filamentPotential>

Sensor Ion Source active Filament Potential as measured on Power Supply board.

type:	integer	\$message:	0x34
version:	0.6.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
units:	mV	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

91. Filament current

</mmsp/diagnosticData/filamentCurrent>

Sensor Ion Source active Filament Current as measured on Power Supply board.

type:	integer	\$message:	0x35
version:	0.6.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
units:	mA	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

92. Electron Multiplier potential

</mmsp/diagnosticData/electronMultiplierPotential>

Sensor Detector Electron Multiplier Potential as measured on Power Supply board.

type:	integer	\$message:	0x37
version:	0.6.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
units:	V	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

93. Measured Data

</mmsp/measurement>

These targets return measured data in two alternative forms.

/scans is an array of scan objects, one of which may be in progress, and where each scan object has its own separate *values* array. The alternative **/data** returns a single object with a single *values* array containing data from any number of scans appended together, the last of which may be in progress.

Each form is described in more detail in the sections below. The advantage of the **/scans** format is less bookkeeping to keep track of boundaries between scans. This form is the simplest for dealing with complete scans. Its disadvantage, however, is when polling for the current scan (e.g. with `measurement/scans/0/get`), because the current scan is incomplete until just before it becomes the prior scan. Polling the current scan repeatedly gets the same data from the beginning of the scan, and could miss the final data when a new scan starts. The advantage the **/data** format, however, is that clients can request only the most current data not yet received, but they must keep track of the boundaries between scans.

type: object \$block: 0x0AXX

94. Total Pressure

[/mmssp/measurement/totalPressure](#)

This target reports the total pressure reading in the sensor.

The emission must be on to get a total pressure reading.

When the emission is off a -1 is returned.

type:	number	\$message:	0x10
version:	0.5.0	\$type:	F32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

95. Versatile Scan Data

[/mmssp/measurement/scans](#)

This target is the primary means to acquire data taken using the Versatile Scan engine. (See also ScansPow2 and ScansBin.) This target is capable of returning large quantities of data depending on the application. Data is returned as an array of objects with the following properties:

scannum	scan count since start of scanning, starting at 1
scansize	Total number of points in a full scan. The actual length of the array of data may be less than this if a partial scan is returned.
values	array of data

The values property is an array of numbers. Note that the numbers may be integer or float, depending on the Versatile Scan setup. It is up to the client to know which type of number to expect. All numbers are padded with spaces to occupy 13 characters. Floats are always in scientific notation (SX.XXXXXXXeSXX), so you can check if the third character is a '.' or the tenth character is an 'e' to determine if it is a float. Inf and Nan values are replaced with -9.999999e-31.

This format may be too slow for some large scans. You may want to use the ScansPow2 message for JavaScript programs, or the ScansBin message for programming languages that can process binary data.

Examples:

/mmssp/measurement/scans/**-1**/get returns the most recently completed scan.

/mmssp/measurement/scans/**-2**/get returns the scan prior to the most recently completed.

/mmssp/measurement/scans/**0**/get returns scan currently in progress.

/mmssp/measurement/scans/**1**/get returns the first scan completed after the scanStart command.

/mmssp/measurement/scans/**1000**/get returns the 1000th scan completed after the scanStart command.

Over time, scan numbers continue to increment, and earlier scans become unavailable. Every available completed scan can be accessed by its absolute scan number, or by a negative number that represents its relative position with respect to the current scan.

type:	array	\$synonyms:	Scans.JSON
version:	0.5.0	\$message:	0x20
minindex:	-100	\$type:	SCAN
readonly:	true	\$flags:	SPECIAL PUBLIC
optional:	true	\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

Array of objects with following properties:

type: object

atomic: true

Note: It is impossible to get or set individual sub-elements of an atomic target.

96. scannum

</mmsp/measurement/scans/<index#>/scannum>

type: integer

readonly: true

97. scansize

</mmsp/measurement/scans/<index#>/scansize>

type: integer

readonly: true

98. values

</mmsp/measurement/scans/<index#>/values>

type:	array
minItems:	0
maxItems:	16384
readonly:	true

Array of number

99. Versatile Scan Data - Pow2 Format

</mmsp/measurement/scansPow2>

This target Returns the versatile scan data in Pow2 format. (See also Scans and ScansBin.) Pow2 is a representation of numbers that is much faster for the MMSP to generate than the standard floating point format. Data is returned as a JSON object with the following properties:

scannum	scan count since start of scanning, starting at 1
scansize	Total number of points in a full scan. The actual length of the array of data may be less than this if a partial scan is returned.
values	array of data

In the Pow2 format, the values property is an array of 1, 2, or 3 element arrays. An integer will be represented by a 1 element array containing the value. A floating point number will be represented by a 2 element array [x,p], where the number is

$$\text{value} = x * 1e-8 * \text{pow}(2.0,p)$$

This format is much faster for the MMSP to generate than the standard floating point format, and can be handled by JavaScript in browsers that do not support directly accessing binary data. Note that $-128 \leq p \leq 128$, so it may make sense to build a lookup table for the $\text{pow}(2.0,p)$. Inf (both positive and negative) is represented as a 3 element array [-1,-100,"inf"] and Nan is represented as [-1,-100,"nan"]

type:	array	\$synonyms:	Scans.Pow2
version:	0.5.0	\$message:	0x23
minindex:	-100	\$type:	SCAN
readonly:	true	\$flags:	SPECIAL PUBLIC
optional:	true	\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

Array of objects with following properties:

type: object

atomic: true

Note: It is impossible to get or set individual sub-elements of an atomic target.

100. scannum

</mmsp/measurement/scansPow2/<index#>/scannum>

type: integer

readonly: true

101. scansize

</mmsp/measurement/scansPow2/<index#>/scansize>

type: integer

readonly: true

102. values

</mmsp/measurement/scansPow2/<index#>/values>

type:	array
minItems:	0
maxItems:	16384
readonly:	true

Note: array items type is non-standard: *array*.

103. Versatile Scan Data - Binary Format

</mmsp/measurement/binaryScans>

Returns the versatile scan in Binary format. The Bin format is a binary header followed by an array of 4-byte binary values (wrapped in an HTTP response).

The binary header is 16 bytes containing the following fields:

0

4-byte	marker	0x0A0B0C0D
1-byte	header_size	Number of 4-byte elements in header, currently 4
1-byte	data_header_size	Number of 4-byte elements in data header, currently 3
2-byte	data_size	Number of 4-byte elements of packet after the data header
2-byte	data_type	0x0053 == 'S'
2-byte	reserved	
4-byte	status	low byte is CCB, upper 3 bytes are reserved, bit 7 - hardware error, bit 6 - hardware warning, bit 5 - comm error , bit 1 - system status changed

The binary header may have fields added to it in the future. The current fields will not be changed. You can always read 16 bytes and check the header_size to determine the offset to the actual data.

After the binary header is the data header. This is 3 4-byte numbers:

4-byte	scansize	number of points in each full scan
4-byte	lastscansize	number of points in the last scan (may be less than the scan size if the last scan is partial)
4-byte	numscans	total number of scans to follow

The data header is followed by some number of scans. Each scan starts with a 2 4-byte numbers header containing:

4-byte	scannum	scan count since start of scanning, starting at 1
4-byte	count	actual number of points in following array. $0 \leq \text{count} \leq \text{scansize}$

This is then followed by 'count' 4-byte numbers (possibly signed or unsigned integers or 32-bit floats. The floats may be nan or inf.)

type:	binaryarray	\$synonyms:	Scans.Bin
version:	0.5.0	\$message:	0x24
minindex:	-100	\$type:	BINSCAN
readonly:	true	\$flags:	SPECIAL PUBLIC
optional:	true	\$readmode:	locked
items:	[object Object]	\$writemode:	null
		\$execmode:	null

Unexpected api type ?binaryarray? binaryScans

104. Versatile Data - JSON Format

</mmsp/measurement/data>

This readonly target returns data from any number of scans within a single array of numbers. (See also DataPow2 and DataBin.) The response is a single JSON object with the following properties:

start	Starting point
scansize	Total number of values per scan
values	array of numbers

Getting this target takes an *@start* parameter, and an optional *@end* parameter. The returned slice of values contains data from position *@start* (indexed by 0) up through and including position *@end*. If *@end* is not specified, all available data beginning with *@start* is returned. The length of the returned values array is thus $(@end - @start) + 1$. The number returned in start is normally equal to the *@start*

parameter if the requested data is available, but if @start is too small the first available position is returned.

Assuming that a previous call responded with scansize:100, and that the first scan is still available, then:
 /mmsp/measurement/data/get?@start=0&@end=99 would return the first scan, and
 /mmsp/measurement/data/get?@start=100&@end=199 would return the second scan.
 /mmsp/measurement/data/get?@start=0 returns all available data.

Numbers in the value array may be integer or float, depending on the scanSetup. It is up to the client to know which type of number to expect. All numbers are padded with spaces to occupy 13 characters. Floats are always in scientific notation (SX.XXXXXXeSXX), so you can check if the third character is a '.' or the tenth character is a 'e' to determine if it is a float. Inf and Nan values are replaced with -9.999999e-31.

This format may be too slow to generate for some large scans. You may want to use the DataPow2 message for JavaScript programs, or the DataBin message for programming languages that can process binary data.

type:	object	\$synonyms:	Data.JSON
version:	0.5.0	\$message:	0x40
readonly:	true	\$type:	DATA
optional:	true	\$flags:	SPECIAL PUBLIC
atomic:	true	\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target.

105. start

</mmsp/measurement/data/start>

type: integer

readonly: true

106. scansize

</mmsp/measurement/data/scansize>

type: integer

readonly: true

107. values
</mmssp/measurement/data/values>

type:	array
minItems:	0
maxItems:	16384
readonly:	true

Array of number

108. Versatile Data - Pow2 Format
</mmssp/measurement/dataPow2>

This target acquires data taken using the Versatile Scan engine. (See also Data and DataBin.) Pow2 is a representation of numbers that is much faster for the MMSP to generate than the standard floating point format.

This target takes a *start* parameter, and an optional *end* parameter. It returns data points starting at *start*. The *start* starts at 0, and is incremented on each data point. If *end* is not specified, all available data starting at *start* is returned. Data is returned as a JSON object with the following properties:

start	Starting point
scansize	Total number of points in a single scan.
values	array of data

In the Pow2 format, the values property is an array of 1, 2, or 3 element arrays. An integer will be represented by a 1 element array containing the value. A floating point number will be represented by a 2 element array [x,p], where the number is

$$\text{value} = x * 1e-8 * \text{pow}(2.0,p)$$

This format is much faster for the MMSP to generate than the standard floating point format, and can be handled by JavaScript in browsers that do not support directly accessing binary data. Note that $-128 \leq p \leq 128$, so it may make sense to build a lookup table for the $\text{pow}(2.0,p)$. Inf (both positive and negative) is represented as a 3 element array [-1,-100,"inf"] and Nan is represented as [-1,-100,"nan"]

type:	object	\$synonyms:	Data.Pow2
version:	0.5.0	\$message:	0x43
readonly:	true	\$type:	DATA
optional:	true	\$flags:	SPECIAL PUBLIC
atomic:	true	\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target.

109. start
</mmssp/measurement/dataPow2/start>

type: integer

readonly: true

110. scansize
</mmssp/measurement/dataPow2/scansize>

type: integer

readonly: true

111. values
</mmssp/measurement/dataPow2/values>

type:	array
minItems:	0
maxItems:	16384
readonly:	true

Note: array items type is non-standard: *array*.**112. Versatile Data - Binary Format**
</mmssp/measurement/binaryData>

This target acquires data taken using the Versatile Scan engine. (See also Data and DataPow2.) This target takes a *start* parameter, and an optional *end* parameter. It returns data points starting at *start*. The *start* starts at 0, and is incremented on each data point. If *end* is not specified, all available data starting at *start* is returned.

The binary header is 16 bytes containing the following fields:

0

4-byte	marker	0x0A0B0C0D
1-byte	header_size	Number of 4-byte elements in header, currently 4
1-byte	data_header_size	Number of 4-byte elements in data header, currently 3
2-byte	data_size	Number of 4-byte elements of packet after the data header
2-byte	data_type	0x44 == 'D'
2-byte	reserved	
4-byte	status	low byte is CCB, upper 3 bytes are reserved, bit 7 - hardware error, bit 6 - hardware warning, bit 5 - comm error , bit 1 - system status changed

The binary header may have fields added to it in the future. The current fields will not be changed. You can always read 16 bytes and check the header_size to determine the offset to the actual data.

After the binary header is the data header. This is 3 4-byte numbers:

4-byte	start	points since start of scanning, starting at 0
4-byte	scansize	Total number of points in a full scan. The size of the array of data may be less than this if a partial scan is returned.
4-byte	count	actual number of points in following array. Count may be larger than scansize.

This is then followed by count 4-byte numbers (possibly signed or unsigned integers or 32-bit floats. The floats may be nan or inf.)

type:	binaryarray	\$synonyms:	Data.Bin
version:	0.5.0	\$message:	0x44
readonly:	true	\$type:	BINDATA
optional:	true	\$flags:	SPECIAL PUBLIC
atomic:	true	\$readmode:	locked
properties:	[object Object]	\$writemode:	null
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target. Unexpected api type ?binaryarray? binaryData

113. Next Scan

</mmsp/measurement/nextScan>

Returns the next scan. This will only return a full scan; if only a partial scan is available then the "values" property will be null.

Each time scanning is started, each session has its next scan number set to 1. Each time NextScan is successfully called, the next scan number is incremented. (It may also be set by the NextScanNumber command.)

The "values" property can have one of the following values:

null	This scan has not yet been completed
An array of values	If the scan is complete and available
[]	If the scan has been completed, but the data is no longer available

The response to nextScan also contains the following properties:

systemStatus	The current value of the systemStatus command
currentScan	The scan number of the scan currently being read
currentScanPoints	The number of points that have been collected in the current scan
scannum	The scan number of this scan
scansize	The number of points in this scan

If NextScan returns "values":null, you can use the "currentScan" and "currentScanPoints" to estimate how long until this scan is complete.

type:	object	\$message:	0x60
version:	0.19.0	\$type:	DATA
readonly:	true	\$flags:	SPECIAL PUBLIC
optional:	true	\$readmode:	locked
atomic:	true	\$writemode:	null
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target.

114. start

</mmsp/measurement/nextScan/start>

type: integer

readonly: true

115. scansize

</mmsp/measurement/nextScan/scansize>

type: integer

readonly: true

116. values

</mmsp/measurement/nextScan/values>

type:	array
minItems:	0
maxItems:	16384
readonly:	true

Array of number

117. Next Scan Number

</mmssp/measurement/nextScanNumber>

Get/Set the scan number used by NextScan. The NextScan Number is kept separately for each session.

type:	integer	\$message:	0x61
version:	0.19.0	\$type:	U32
		\$flags:	SPECIAL PUBLIC
		\$readmode:	locked
		\$writemode:	locked
		\$execmode:	null

118. Binary Next Scan

</mmssp/measurement/binaryNextScan>

Returns the next scan. This will only return a full scan; if only a partial scan is available then the data will be empty.

Each time scanning is started, each session has its next scan number set to 1. Each time BinaryNextScan is successfully called, the next scan number is incremented. (It may also be set by the NextScanNumber command.)

The response contains the standard 16-byte binary header, followed by a 24 byte data header, possibly followed by an array of data.

The binary header is 16 bytes containing the following fields:

0

4-byte	marker	0x0A0B0C0D
1-byte	header_size	Number of 4-byte elements in header, currently 4
1-byte	data_header_size	Number of 4-byte elements in data header, currently 3
2-byte	data_size	Number of 4-byte elements of packet after the data header
2-byte	data_type	0x4E == 'N'
2-byte	reserved	
4-byte	status	low byte is CCB, upper 3 bytes are reserved, bit 7 - hardware error, bit 6 - hardware warning, bit 5 - comm error , bit 1 - system status changed

The binary header may have fields added to it in the future. The current fields will not be changed. You can always read 16 bytes and check the header_size to determine the offset to the actual data.

After the binary header is the data header. This is 5 4-byte numbers:

4-byte	systemStatus	Points since start of scanning, starting at 0
4-byte	curScan	Total number of points in a full scan. The size of the array of data may be less than this if a partial scan is returned.
4-byte	curScanPoints	Total number of points in a full scan. The size of the array of data may be less than this if a partial scan is returned.
4-byte	npoints	Actual number of points in following array. Will either be 0 or scansize
4-byte	scannum	The scan number of this scan
4-byte	scansize	Total number of points in a complete scan

If npoints is greater than 0, this will be followed by npoints 4-byte values (may be unsigned, signed or float, depending on channel definition.)

type:	binaryarray	\$message:	0x64
version:	0.17.0	\$type:	BINDATA
readonly:	true	\$flags:	SPECIAL PUBLIC
optional:	true	\$readmode:	locked
atomic:	true	\$writemode:	null
properties:	[object Object]	\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target. Unexpected api type ?binaryarray? binaryNextScan

119. Scan Info

[/mmssp/scanInfo](#)

Returns the following information about the current scans

type: object \$block: 0x0B0X

120. First Scan

[/mmssp/scanInfo/firstScan](#)

Scan Number of first complete scan still available in scan queue, or -1 if no complete scans are available.

type:	integer	\$message:	0x10
version:	0.9.0	\$type:	U32
readonly:	true	\$flags:	SPECIAL PUBLIC
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

121. Last Scan

[/mmssp/scanInfo/lastScan](#)

Scan Number of last complete scan still available in scan queue, or -1 if no complete scans are available.

type:	integer	\$message:	0x11
version:	0.9.0	\$type:	U32
readonly:	true	\$flags:	SPECIAL PUBLIC
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

122. Current Scan

[/mmssp/scanInfo/currentScan](#)

Scan Number of the current scan, or -1 if no complete scans are available.

type:	integer	\$message:	0x12
version:	0.9.0	\$type:	U32
readonly:	true	\$flags:	SPECIAL PUBLIC
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

123. Points Per Scan

[/mmsp/scanInfo/pointsPerScan](#)

Total number of points in a complete scan.

type:	integer	\$message:	0x13
version:	0.9.0	\$type:	U32
readonly:	true	\$flags:	SPECIAL PUBLIC
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

124. Points In Current Scan

[/mmsp/scanInfo/pointsInCurrentScan](#)

Number of points currently read into current scan.

type:	integer	\$message:	0x14
version:	0.9.0	\$type:	U32
readonly:	true	\$flags:	SPECIAL PUBLIC
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

125. Scanning

[/mmsp/scanInfo/scanning](#)

True if scanning is currently running.

type:	string	\$message:	0x15
version:	0.9.0	\$type:	U8
readonly:	true	\$flags:	SPECIAL PUBLIC
enum:	False,True	\$range:	False,True
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

126. Sensor Detector parameters

[/mmsp/sensorDetector](#)

type: object \$block: 0x17XX

127. EM Voltage Max

[/mmsp/sensorDetector/emVoltageMax](#)

Get electron multiplier high boundary in volts.

type:	integer	\$message:	0x20
version:	0.5.0	\$type:	S32
readonly:	true	\$emul:	2000
units:	V	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

128. EM Voltage Min

[/mmsp/sensorDetector/emVoltageMin](#)

Get electron multiplier low boundary in volts.

type:	integer	\$message:	0x21
version:	0.5.0	\$type:	U32
readonly:	true	\$emul:	600
units:	V	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

129. EM Voltage

[/mmsp/sensorDetector/emVoltage](#)

Get or set the working electron multiplier voltage.

type:	integer	\$message:	0x30
version:	0.5.0	\$type:	S32
units:	V	\$emul:	800
		\$flags:	PUBLIC
		\$range:	EMVoltageMin:EMVoltageMax
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

130. EM Gain

[/mmsp/sensorDetector/emGain](#)

Get or set the gain at the set electron multiplier voltage as calibrated.

type:	number	\$message:	0x31
version:	1.2.0	\$type:	F32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

131. EM Gain Mass

[/mmsp/sensorDetector/emGainMass](#)

Get or Set the mass in cAMU where the EM gain is calculated.

type:	integer	\$message:	0x32
version:	1.2.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

132. Load Default Parameters

[/mmsp/sensorDetector/defaultLoad](#)

This target loads the working detector parameters from the default nominal values.

type:	integer	\$message:	0x34
version:	0.9.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

133. Load Factory Parameters

[/mmssp/sensorDetector/factoryLoad](#)

This target loads the working detector parameters from the factory nonvolatile area and makes them operational.

type:	integer	\$message:	0x35
version:	0.9.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

134. Save Factory Parameters

[/mmssp/sensorDetector/factorySave](#)

This target saves the working detector parameters to factory nonvolatile memory.

type:	integer	\$message:	0x36
version:	0.9.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

135. Load User Parameters

[/mmssp/sensorDetector/userLoad](#)

This target loads the working detector parameters from the user nonvolatile area and makes them operational.

type:	integer	\$message:	0x37
version:	0.9.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

136. Save User Parameters

[/mmsp/sensorDetector/userSave](#)

This target saves the working detector parameters to user nonvolatile memory.

type:	integer	\$message:	0x38
version:	0.9.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

137. Sensor Ion Source parameters

[/mmsp/sensorIonSource](#)

type: object \$block: 0x18XX

138. Max Emission Current

[/mmsp/sensorIonSource/emissionCurrentMax](#)

The maximum user settable emission current in microampere (uA).

type:	integer	\$message:	0x20
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	50
units:	uA	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

139. Min Emission Current

[/mmsp/sensorIonSource/emissionCurrentMin](#)

The minimum user settable emission current in microampere (uA).

type:	integer	\$message:	0x21
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	0
units:	uA	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

140. Anode Voltage Upper Bound

</mmssp/sensorIonSource/anodeMax>

The maximum user settable anode potential in Volts (V).

type:	integer	\$message:	0x22
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	212
units:	V	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

141. Anode Voltage Lower Bound

</mmssp/sensorIonSource/anodeMin>

The minimum user settable anode potential in Volts (V).

type:	integer	\$message:	0x23
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	80
units:	V	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

142. Max Focus Voltage

</mmssp/sensorIonSource/focusMax>

The maximum user settable focus plate potential, relative to anode in Volts (V).

type:	integer	\$message:	0x24
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	200
units:	V	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

143. Min Focus Voltage

</mmsp/sensorIonSource/focusMin>

The minimum user settable focus potential, relative to anode in Volts (V).

type:	integer	\$message:	0x25
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	0
units:	V	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

144. Electron Energy Upper Bound

</mmsp/sensorIonSource/electronEnergyMax>

The maximum user settable Electron Energy potential in Volts (V).

type:	integer	\$message:	0x26
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	100
units:	V	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

145. Electron Energy Lower Bound

</mmsp/sensorIonSource/electronEnergyMin>

The minimum user settable Electron Energy potential in Volts (V).

type:	integer	\$message:	0x27
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	10
units:	V	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

146. Ion Energy Max

</mmsp/sensorIonSource/ionEnergyMax>

Largest ion energy value in meV (millielectronvolts) which the user can set.

type:	integer	\$message:	0x28
version:	0.7.0	\$type:	U16
readonly:	true	\$flags:	CNST PUBLIC
units:	mV	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

147. Ion Energy Min

</mmsp/sensorIonSource/ionEnergyMin>

Smallest ion energy value in meV (millielectronvolts) which the user can set.

type:	integer	\$message:	0x29
version:	0.7.0	\$type:	U16
readonly:	true	\$flags:	CNST PUBLIC
units:	mV	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

148. Structures Max

</mmsp/sensorIonSource/structuresMax>

The maximum number of ion source operating points definable.

type:	integer	\$message:	0x2C
version:	0.5.0	\$type:	U16
readonly:	true	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

149. Activate

</mmssp/sensorIonSource/activate>

This target makes working Ion Source parameters operational.

type:	integer	\$message:	0x39
version:	0.5.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

150. Selected Filament

</mmssp/sensorIonSource/filamentSelected>

Get or Set the currently selected filament.

Setting this target will also issue a load of the working ion source from the user table specified by selected filament, emission level and optimization type.

Getting this target will return the currently selected filament with no working ion source load.

Possible values are 1 and 2.

type:	integer	\$message:	0x3A
version:	0.5.0	\$type:	U8
minimum:	1	\$emul:	1
maximum:	2	\$flags:	PUBLIC
		\$range:	1:2
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

151. Selected Filament Override

</mmssp/sensorIonSource/filamentSelectedOverride>

Get or Set the selected filament without changing the ion source parameters.

Possible values are 1 and 2.

type:	integer	\$message:	0x3B
version:	0.5.0	\$type:	U8
minimum:	1	\$flags:	PUBLIC
maximum:	2	\$range:	1:2
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

152. Working Structure

</mmssp/sensorIonSource/workingStructure>

Sensor Ion Source Targets Full View

Sensor Ion Source Targets Working Expanded View

This target allows reading or writing the working Mass Filter parameters.

type:	string	\$message:	0x3C
version:	0.5.0	\$type:	BINARY
atomic:	true	\$emul:	1,2,3,4,5,6,7,8,9,0
		\$flags:	PUBLIC
		\$length:	10
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target.

153. Emission Level

</mmssp/sensorIonSource/emissionLevel>

Get or Set the emission level (Low or High) selector for the working ion source values.

type:	string	\$message:	0x40
version:	0.5.0	\$type:	U8
enum:	Lo,Hi	\$flags:	PUBLIC
		\$range:	Lo,Hi
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

154. Optimization Type

[/mmssp/sensorIonSource/optimizationType](#)

Get or Set the optimization type (Linearity or Sensitivity) selector for the working ion source values.

type:	string	\$message:	0x41
version:	0.5.0	\$type:	U8
enum:	Linearity,Sensitivity	\$flags:	PUBLIC
		\$range:	Linearity,Sensitivity
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

155. Filament Preferred

[/mmssp/sensorIonSource/filamentPreferred](#)

Get or Set the ion source preferred filament as determined when tuned.

Possible values are 1 and 2.

type:	integer	\$message:	0x42
version:	0.5.0	\$type:	U8
minimum:	0	\$flags:	PUBLIC
maximum:	2	\$range:	0:2
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

156. Nominal To User Copy

[/mmssp/sensorIonSource/nomUserCopy](#)

This target copies the default (nominal values) to the user ion source table. The value must be set to 123.

type:	integer	\$message:	0x48
version:	0.62.0	\$type:	U16
writeonly:	true	\$flags:	PUBLIC
optional:	true	\$readmode:	null
		\$writemode:	control
		\$execmode:	null

157. Nominal to Factory Copy

[/mmssp/sensorIonSource/nomFactoryCopy](#)

This target copies the default (nominal values) to factory ion source table. The value must be set to 234.

type:	integer	\$message:	0x49
version:	0.62.0	\$type:	U16
writeonly:	true	\$flags:	PUBLIC
optional:	true	\$readmode:	null
		\$writemode:	control
		\$execmode:	null

158. User Tables Valid

[/mmssp/sensorIonSource/userTablesValid](#)

Get the status of all ion source user tables. A single bit is used to indicate if a table is valid.

Bit 0 when set indicates table 1 is valid.

Bit 1 when set indicates table 2 is valid.

.....

Bit 7 when set indicates table 8 is valid.

type:	integer	\$message:	0x4A
version:	0.62.0	\$type:	U16
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

159. Factory Tables Valid

[/mmssp/sensorIonSource/factoryTablesValid](#)

Get the status of all ion source factory tables. A single bit is used to indicate if a table is valid.

Bit 0 when set indicates table 1 is valid.

Bit 1 when set indicates table 2 is valid.

.....

Bit 7 when set indicates table 8 is valid.

type:	integer	\$message:	0x4B
version:	0.62.0	\$type:	U16
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

160. Working Load Source

</mmsp/sensorIonSource/workingLoadSrc>

Get the load source for the current working tune.

1 = default

2 = factory

3 = user

4 = download

type:	integer	\$message:	0x4C
version:	0.62.0	\$type:	U16
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

161. Emission current

</mmsp/sensorIonSource/emissionCurrent>

Get or Set the working ion source emission current in microampere (uA).

type:	integer	\$message:	0x50
version:	0.5.0	\$type:	S32
units:	uA	\$emul:	5500
		\$flags:	PUBLIC
		\$range:	EmissionCurrentMin:EmissionCurrentMax
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

162. Anode Voltage

</mmsp/sensorIonSource/anode>

Get or Set the working ion source anode potential in Volts (V).

type:	integer	\$message:	0x51
version:	0.5.0	\$type:	S32
units:	V	\$emul:	80
		\$flags:	PUBLIC
		\$range:	AnodeMin:AnodeMax
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

163. Focus Voltage

</mmsp/sensorIonSource/focus>

Get or Set the working ion source focus potential in Volts (V).

type:	integer	\$message:	0x52
version:	0.5.0	\$type:	S32
units:	V	\$emul:	65
		\$flags:	PUBLIC
		\$range:	FocusMin:FocusMax
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

164. Electron Energy

</mmsp/sensorIonSource/electronEnergy>

Get or Set the working ion source electron energy potential in millivolts (mV).

type:	integer	\$message:	0x53
version:	0.5.0	\$type:	S32
units:	V	\$emul:	70
		\$flags:	PUBLIC
		\$range:	ElectronEnergyMin:ElectronEnergyMax
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

165. Ion Energy Global

</mmsp/sensorIonSource/ionEnergyGlobal>

Get or Set the working ion source ion energy potential in millivolts (mV). When non-zero, this value will set the sensor to a static ion energy, independent of mass being measured.

type:	integer	\$message:	0x54
version:	0.7.0	\$type:	S32
units:	mV	\$flags:	PUBLIC
		\$range:	IonEnergyMin:IonEnergyMax
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

166. Normalization Factor

[/mmsp/sensorIonSource/normalizationFactor](#)

Get or Set the working ion source normalization factor.

type:	number	\$message:	0x55
version:	0.5.0	\$type:	F32
		\$emul:	70
		\$flags:	PUBLIC
		\$range:	-1e12:1e12
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

167. PP Linearization Constant 1

[/mmsp/sensorIonSource/ppLinConst1](#)

Get or Set the working ion source partial pressure linearization equation constant 1.

type:	number	\$message:	0x65
version:	0.15.0	\$type:	F32
		\$flags:	NVM PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

168. PP Linearization Constant 2

[/mmsp/sensorIonSource/ppLinConst2](#)

Get or Set the working ion source partial pressure linearization equation constant 2.

type:	number	\$message:	0x66
version:	0.15.0	\$type:	F32
		\$flags:	NVM PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

169. PP Sensitivity Factor

</mmssp/sensorIonSource/ppSensitivityFactor>

Get or Set the working ion source partial pressure sensitivity factor in Ampere per calibrated pressure unit.

type:	number	\$message:	0x67
version:	0.17.0	\$type:	F32
		\$flags:	NVM PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

170. Total Pressure Emission Trip Threshold Current

</mmssp/sensorIonSource/tPThresholdCurrent>

Get or Set the ion source total pressure Emission trip threshold current in ampere.

type:	number	\$message:	0x70
version:	0.17.0	\$type:	F32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

171. Total Pressure Emission Trip Threshold Count

</mmssp/sensorIonSource/tPTresholdCount>

Get or Set the ion source total pressure trip treshold count.

This is the number of extra times the total pressure current must exceed the threshold current to constitute a trip. Each count adds approximatly 10ms to the time required for the current to be above the threshold. Settable values 0-65535.

type:	integer	\$message:	0x71
version:	0.17.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

172. Total Pressure Emission Trip Count

</mmsp/sensorIonSource/tPTripCount>

Get or Set the number of times the total pressure current caused the emission to be turned off.

type:	integer	\$message:	0x72
version:	0.17.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

173. Total Pressure EM Trip Threshold Current

</mmsp/sensorIonSource/tPEMThresholdCurrent>

Get or Set the ion source total pressure EM trip threshold current in ampere.

type:	number	\$message:	0x74
version:	1.4.2	\$type:	F32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

174. Total Pressure EM Trip Threshold Count

</mmsp/sensorIonSource/tPEMThresholdCount>

Get or Set the ion source EM total pressure trip threshold count.

This is the number of extra times the total pressure current must exceed the threshold current to constitute a trip. Each count adds approximately 10ms to the time required for the current to be above the threshold. Settable values 0-65535.

type:	integer	\$message:	0x75
version:	1.4.2	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

175. TP EM Trip Count

</mmsp/sensorIonSource/tPEMTripCount>

Get or Set the number of times the total pressure current caused the Electron Multiplier to be turned off.

type:	integer	\$message:	0x76
version:	1.4.2	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

176. Total Pressure Calibration Points Max

</mmsp/sensorIonSource/tPCalPointsMax>

Maximum number of settable ion source total pressure calibration points.

type:	integer	\$message:	0x80
version:	0.17.0	\$type:	U16
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

177. Total Pressure Calibration Points Min

</mmsp/sensorIonSource/tPCalPointsMin>

Minimum number of settable ion source total pressure calibration points required.

type:	integer	\$message:	0x81
version:	0.17.0	\$type:	U16
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

178. Total Pressure Calibration Points in Use

[/mmsp/sensorIonSource/tPCalPointsInUse](#)

Get or Set the number of ion source total pressure calibration points to use.

type:	integer	\$message:	0x82
version:	0.17.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

179. Total Pressure Calibration Pressure Set

[/mmsp/sensorIonSource/tPCalPressSet](#)

Set an ion source total pressure calibration pressure to add or update. Issue this target when the ion source is at the desired pressure, as the current associated with the pressure will be captured. If the specified pressure is within +/- 10% of an existing point the pressure value will be updated. If the specified pressure is equal or outside a +/- 10% window of an existing point a new calibration point will be added and /mmsp/sensorIonSource/tPCalPointsInUse will be incremented. A Get will return the last specified update or add pressure.

type:	number	\$message:	0x85
version:	0.17.0	\$type:	F32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

180. Total Pressure Calibration Pressure Remove

[/mmsp/sensorIonSource/tPCalPressRemove](#)

Set an ion source total pressure calibration point to remove by specified pressure. This function will remove a found calibration point with a pressure within +/- 10% of the specified pressure. If successful the /mmsp/sensorIonSource/tPCalPointsInUse will be decremented. A Get will return the last specified remove pressure.

type:	number	\$message:	0x87
version:	0.17.0	\$type:	F32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

181. Total Pressure Calibration Point Remove

[/mmsp/sensorIonSource/tPCalPointRemove](#)

Set an ion source total pressure calibration point to remove by specified number. If successful the /mmsp/sensorIonSource/tPCalPointsInUse will be decremented. A Get will return the pressure of the last specified remove point.

type:	integer	\$message:	0x88
version:	0.17.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

182. Total Pressure Calibration Table Shift

[/mmsp/sensorIonSource/tPCalTableShift](#)

Set with a known pressure will cause the existing calibration table to be shifted to align with the given pressure. A Get will return the last specified shift pressure (0.0e0 if never set).

Note: This function is best performed with a pressure in the middle of the RGA's operating range (1e-6 to 1e-5 Torr or equivalent).

type:	number	\$message:	0x89
version:	0.47.0	\$type:	F32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

183. TP units

[/mmsp/sensorIonSource/tPunits](#)

Get or Set the ion source total pressure reporting units (0=Torr, 1=mbar, 2=Pascal).

type:	integer	\$message:	0x8C
version:	0.47.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

184. Degas Duration

</mmssp/sensorIonSource/degasDuration>

Get or Set the degas duration in seconds (s).

type:	integer	\$message:	0xA8
version:	0.47.0	\$type:	U16
		\$flags:	
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

185. Emis Restart Delay

</mmssp/sensorIonSource/emisRestartDelay>

Set or Get the Emission/EM restart delay time after an emission trip for over pressure, emission error or anode error. Settable values are 0-65535 seconds. A set value of zero will disable the restart function.

type:	integer	\$message:	0x90
version:	0.47.0	\$type:	U16
units:	sec	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

186. Emis Restart Count

</mmssp/sensorIonSource/emisRestartCount>

Set or Get the Emission/EM restart try count after an emission trip for over pressure, emission error or anode error. Settable values are 0-255. A set value of zero disables the function.

type:	integer	\$message:	0x91
version:	0.47.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

187. Emis Restart Type

</mmsp/sensorIonSource/emisRestartType>

Set or Get the Emission/EM restart type. Settable values are 0 for Emission restart and 1 for Emission and EM.

type:	integer	\$message:	0x92
version:	0.47.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

188. calIndex

</mmsp/sensorIonSource/calIndex>

type:	array	\$range:	0:10
minIndex:	0		
minItems:	0		
maxItems:	10		

Array of objects with following properties:

type: object

189. Total Pressure Calibration Point Pressure

</mmsp/sensorIonSource/calIndex/<index#>/tPCalPointPress>

Get or Set an ion source total pressure calibration point pressure by specified number.

type:	number	\$message:	0x83
version:	0.37.0	\$type:	F32
		\$index:	CalIndex
		\$flags:	PUBLIC
		\$length:	0:10
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

190. Total Pressure Calibration Point Current

[/mmsp/sensorIonSource/calIndex/<index#>/tPCalPointCurrent](#)

Get or Set an ion source total pressure calibration point current in Ampere by specified number.

type:	number	\$message:	0x84
version:	0.37.0	\$type:	F32
		\$index:	CalIndex
		\$flags:	PUBLIC
		\$length:	0:10
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

191. Total Pressure Calibration Point Set

[/mmsp/sensorIonSource/calIndex/<index#>/tPCalPointSet](#)

Set an ion source total pressure calibration point. Issue this target when the ion source is at the desired pressure, as the current associated with the pressure will be captured.

Get will return the last specified pressure for a given point.

type:	number	\$message:	0x86
version:	0.47.0	\$type:	F32
		\$index:	CalIndex
		\$flags:	PUBLIC
		\$length:	0:10
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

192. ionSource

</mmssp/sensorIonSource/ionSource>

type:	array	\$range:	0:8
minIndex:	0		

Array of objects with following properties:

type: object

193. Load Default Parameters

</mmssp/sensorIonSource/ionSource/<index#>/defaultLoad>

This target loads the working ion source parameters from the default nominal values.

type:	integer	\$message:	0x34
version:	0.5.0	\$type:	CMD
exec:	true	\$index:	IonSource
writeonly:	true	\$flags:	PUBLIC
optional:	true	\$length:	0:8
		\$readmode:	null
		\$writemode:	null
		\$execmode:	control

194. Load Factory Parameters

</mmssp/sensorIonSource/ionSource/<index#>/factoryLoad>

This target loads the working Ion Source parameters from the factory nonvolatile memory and makes them operational.

The selection of which factory tune structure to read is based on the sent value as follows:

0 = Read from the structure selected by the current selected filament, optimization type and emission level.

1-8 = Read from the first factory structure.

type:	integer	\$message:	0x35
version:	0.5.0	\$type:	CMD
exec:	true	\$index:	IonSource
writeonly:	true	\$flags:	PUBLIC
optional:	true	\$length:	0:8
		\$readmode:	null
		\$writemode:	null
		\$execmode:	control

195. Load User Parameters

</mmsp/sensorIonSource/ionSource/<index#>/userLoad>

This target loads the working Ion Source parameters from the user nonvolatile memory and makes them operational.

The selection of which user structure to read is based on the sent value as follows:

0 = Read from the structure selected by the current selected filament, optimization type and emission level.

1-8 = Read from the numbered user structure.

type:	integer	\$message:	0x37
version:	0.5.0	\$type:	CMD
exec:	true	\$index:	IonSource
writeonly:	true	\$flags:	PUBLIC
optional:	true	\$length:	0:8
		\$readmode:	null
		\$writemode:	null
		\$execmode:	control

196. Save User Parameters

</mmsp/sensorIonSource/ionSource/<index#>/userSave>

This target saves working Ion Source parameters to user nonvolatile memory.

The selection of which user structure to write to is based on the sent value as follows:

0 = Write to the structure selected by the current selected filament, optimization type and emission level.

1-8 = Write to the numbered user structure.

type:	integer	\$message:	0x38
version:	0.5.0	\$type:	CMD
exec:	true	\$index:	IonSource
writeonly:	true	\$flags:	PUBLIC
optional:	true	\$length:	0:8
		\$readmode:	null
		\$writemode:	null
		\$execmode:	control

197. Sensor Filter Parameters

[/mmsp/sensorFilter](#)

type: object \$block: 0x19XX

198. Tune Max Mass

[/mmsp/sensorFilter/massMax](#)

Maximum mass in AMU that the user can schedule for a measurement.

type:	number	\$baseunits:	cAMU
version:	0.5.0	\$message:	0x10
readonly:	true	\$type:	U16
divisibleBy:	0.01	\$emul:	30000
units:	AMU	\$default:	300.00
		\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

199. Tune Min Mass

[/mmsp/sensorFilter/massMin](#)

Minimum mass in AMU that the user can schedule for a measurement.

type:	number	\$baseunits:	cAMU
version:	0.5.0	\$message:	0x11
readonly:	true	\$type:	U16
divisibleBy:	0.01	\$emul:	100
units:	AMU	\$default:	1.00
		\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

200. Max Allowable Dwell

[/mmssp/sensorFilter/dwellMax](#)

Returns the maximum dwell supported by this device.

type:	number	\$baseunits:	usec
version:	1.0.2	\$message:	0x18
readonly:	true	\$type:	U32
divisibleBy:	0.001	\$flags:	SPECIAL PUBLIC
units:	msec	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

201. Min Allowable Dwell

[/mmssp/sensorFilter/dwellMin](#)

Returns the minimum dwell supported by this device.

type:	number	\$baseunits:	usec
version:	1.0.2	\$message:	0x19
readonly:	true	\$type:	U32
divisibleBy:	0.001	\$flags:	SPECIAL PUBLIC
units:	msec	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

202. Design Max Mass

[/mmssp/sensorFilter/designMassMax](#)

Minimum mass in AMU that the user can schedule for a measurement.

type:	number	\$baseunits:	cAMU
version:	1.1.0	\$message:	0x38
readonly:	true	\$type:	U16
divisibleBy:	0.01	\$emul:	330.00
units:	AMU	\$default:	330.00
		\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

203. Design Min Mass

[/mmsp/sensorFilter/designMassMin](#)

Minimum mass in AMU that the user can schedule for a measurement.

type:	number	\$baseunits:	cAMU
version:	1.1.0	\$message:	0x39
readonly:	true	\$type:	U16
divisibleBy:	0.01	\$emul:	100
units:	AMU	\$default:	1.00
		\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

204. Rod Polarity

[/mmsp/sensorFilter/rodPolarity](#)

Get or Set mass filter rod polarity and load mass filter operating parameters based on polarity. May be either Normal(1) or Reverse(2).

type:	integer	\$message:	0x40
version:	0.5.0	\$type:	U8
minimum:	0	\$flags:	PUBLIC
maximum:	2	\$range:	0:2
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

205. Working Tune

[/mmsp/sensorFilter/workingTune](#)

Get or Set the working Tune structure.

type:	string	\$message:	0x65
version:	0.5.0	\$type:	BINARY
atomic:	true	\$emul:	1,2,3,4,5,6,7,8,9,0
		\$flags:	PUBLIC
		\$length:	10
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

Note: It is impossible to get or set individual sub-elements of an atomic target.

206. Max Allowable Dwell (fixed)

</mmssp/sensorFilter/maxDwell>

Returns the maximum dwell supported by this device.

type:	number	\$baseunits:	usec
version:	1.1.0	\$message:	0xf8
readonly:	true	\$type:	U32
divisibleBy:	0.001	\$flags:	SPECIAL PUBLIC
units:	msec	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

207. Min Allowable Dwell (fixed)

</mmssp/sensorFilter/minDwell>

Returns the minimum dwell supported by this device.

type:	number	\$baseunits:	usec
version:	1.1.0	\$message:	0xf9
readonly:	true	\$type:	U32
divisibleBy:	0.001	\$flags:	SPECIAL PUBLIC
units:	msec	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

208. Versatile Scan Parameters

</mmssp/scanSetup>

Versatile Scan overview:

A Versatile scan is comprised of 300 general-purpose available channels. Each channel may be a single reported value (single mass measurement or an area under a sweep), an entire sweep (start, end & points per amu), or a special value (Calendar time, timer value, internal value, etc.)

The scanSetup and channel settings of this target determine the values that will appear in /measurement once scanning has been started. See the API Guide for examples of how to set up scanSetup for some common applications.

Note: Before making any changes to scanSetup/channels, scanning must be stopped with scanSetup/scanStop. It can be resumed later with scanSetup/scanStart. Making changes to scanSetup/channels while scanning produces invalid data.

Single reported values: (not inclusive)

A single mass partial pressure reading

A single mass current reading

A single mass relative reading

An area under a sweep of masses
 The ion gauge total pressure reading
 The Pirani total pressure reading
 RF only total pressure reading
 Calendar time (milliseconds since 1 Jan 1900)
 One of two user loadable 32 bit timer values
 One of many internal values (TBD)

Multiple reported values:

A sweep of multiple masses (integer or fractional) partial pressures
 A sweep of multiple masses (integer or fractional) currents
 A sweep of multiple masses (integer or fractional) relative values
 Sweeps are defined by start, stop, points per amu, dwell, lead-in and extra parameters
 Quadrupole sensor (Partial pressure) collection
 Choice of Faraday cup or Electron multiplier detector
 Choice of electron current or equivalent (electron current/multiplier gain) ion current when Electron multiplier on
 Scaling of channel data by a fixed multiplier

Each channel's returned data may be operated on individually by user defined functions
 Scan control for starting, stopping (on scan boundaries) & aborting (not on scan boundaries). The scan proceeds from the startChannel to the stopChannel then back to the startChannel.

Started by

Communications link command

Events (external inputs, Calendar time, timers, etc.)

Stopped by

Communications link command

Events (external inputs, Calendar time, timers, etc.)

Aborted by

Communications link command

Events (external inputs, Calendar time, timers, etc.)

These targets define what to measure. See message block 0x0AXX to get the resulting data.

type: object \$block: 0x1CXX

209. Versatile Max Scan Channels

</mmssp/scanSetup/maxChannels>

This target queries the number of Versatile scan channels available.

type:	integer	\$message:	0x10
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	340
		\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

210. Scan Start Channel

[/mmssp/scanSetup/startChannel](#)

This target queries or updates the starting channel (inclusive) of the Versatile scan.

For a channel to be used, it must fall within the startChannel to stopChannel range AND it must be enabled.

type:	integer	\$message:	0x11
version:	0.5.0	\$type:	U16
		\$emul:	1
		\$flags:	PUBLIC
		\$range:	1:MaxChannels
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

211. Scan Stop Channel

[/mmssp/scanSetup/stopChannel](#)

This target queries or updates the stopping channel (inclusive) of the Versatile scan.

For a channel to be used, it must fall within the startChannel to stopChannel range AND it must be enabled.

type:	integer	\$message:	0x12
version:	0.5.0	\$type:	U16
		\$emul:	1
		\$flags:	PUBLIC
		\$range:	1:MaxChannels
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

212. Scan Count

[/mmssp/scanSetup/scanCount](#)

This target queries or updates the scan count.

If a schedule is not running:

Setting the count between 1-1000 causes that number of scans to be scheduled.

Setting the count to -1 causes an infinite number of scans to be scheduled.

If a schedule is running:

Setting the count between 1-1000 reloads the count to the new value.

Setting the count to -1 causes an infinite number of scans to be run.

type:	integer	\$message:	0x40
version:	0.5.0	\$type:	S16
		\$flags:	PUBLIC
		\$length:	-1:1000
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

213. Scan Start

[/mmssp/scanSetup/scanStart](#)

Starts scanning, if not already running scanning, using current scanSetup.

type:	integer	\$message:	0x41
version:	0.5.0	\$type:	U8
writeonly:	true	\$flags:	PUBLIC
optional:	true	\$readmode:	null
		\$writemode:	control
		\$execmode:	null

214. Scan Stop

[/mmssp/scanSetup/scanStop](#)

If 0, stop scanning at end of current scan. ScanSetup remains unchanged.

If 1, stops scanning immediately.

type:	string	\$message:	0x42
version:	0.5.0	\$type:	U16
enum:	EndOfScan,Immediately	\$flags:	PUBLIC
		\$range:	EndOfScan,Immediately
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

215. Scan Interval

[/mmssp/scanSetup/scanInterval](#)

Set or Get the interval between the start of scans in milliseconds.

When set to zero, scans will occur head to tail with no delay between.

When set to a value less than ScanTimeTotal, the interval will be adjusted to zero.

When set to a value greater than ScanTimeTotal and less than ScanTimeTotal+3ms, the interval will be adjusted to ScanTimeTotal+3ms.

Range of values 0, 5-1000000000 milliseconds.

type:	integer	\$message:	0x43
version:	0.5.1	\$type:	U32
units:	msec	\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

216. Dwell Global

[/mmssp/scanSetup/dwellGlobal](#)

This value when non-zero will override the channel set dwells for measurements using the sensor filter.

This value may be set to zero to turn off or 1 to 16384 milliseconds. This value update will be synchronized to the start of a scan.

type:	number	\$baseunits:	usec
version:	0.5.1	\$message:	0x44
divisibleBy:	0.001	\$type:	U32
units:	msec	\$flags:	PUBLIC
		\$range:	0:/sensorFilter/maxDwell
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

217. Lead In Dwell

[/mmssp/scanSetup/leadInDwell](#)

Set or Get the non-reportable Versiscan lead in dwell in microseconds.

type:	number	\$baseunits:	usec
version:	0.9.0	\$message:	0x45
divisibleBy:	0.001	\$type:	U32
units:	msec	\$flags:	PUBLIC
		\$range:	/sensorFilter/minDwell:/sensorFilter/maxDwell
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

218. Leak Check Mass

[/mmssp/scanSetup/leakCheckMass](#)

Get or Set the value of leak check mass in AMU.

type:	integer	\$message:	0xF0
version:	0.17.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

219. Scan Time Total

[/mmssp/scanSetup/scanTimeTotal](#)

Average time required to complete any scan subsequent to the first scan after a start is issued. Includes time to acquire reportable measurements, baseline measurements if enabled and average peakfind time if enabled. To get the first scan time use "ScanTimeTotalFirstOne".

type:	number	\$baseunits:	usec
version:	0.8.0	\$message:	0x80
readonly:	true	\$type:	U32
divisibleBy:	0.001	\$flags:	PUBLIC
units:	msec	\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

220. channel
</mmssp/scanSetup/channel>

type:	array	\$range:	1:MaxChannels
minIndex:	1	\$active:	StartChannel:StopChannel

Array of objects with following properties:

(Note this array begins with index **1**, so indexing by 0 is invalid)

type: object

221. Channel Type
</mmssp/scanSetup/channel/<index#>/channelType>

Do Not Use This Command

Use ChannelMode instead. This ChannelType command is a low level command that is set automatically by ChannelMode.

This target queries or updates the Channel type. Available types are:

1 = Sweep of points

2 = Single point

3 = Baseline

4 = TPQuad

6 = cAMU User

7 = DAC User

8 = Non Quad U32 returned Special Value

9 = Non Quad F32 returned Special Value

type:	string	\$message:	0x13
version:	0.5.0	\$type:	U16
enum:	Sweep,Single, Baseline,TPQuad, camuUser,dacUser, U32Value,F32Value	\$emul:	1
		\$index:	Channel
		\$flags:	PUBLIC
		\$range:	Sweep,Single, Baseline,TPQuad, camuUser,dacUser, U32Value,F32Value
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

222. Raw Channel # Start Mass or flag

[/mmsp/scanSetup/channel/<index#>/startMassRaw](#)

Do Not Use This Command

Use StartMass instead. This StartMassRaw command is a low level command that is set automatically by StartMass and ChannelMode.

0-30000,special value1

This target queries or updates channel #'s starting mass or special value1.

The starting mass is in centiAMU. The special values are Channel Type dependent.

For Type 6 (cAMU User) Special value1 sets the RF equivalent mass value.

For Type 7 (DAC User) Special value1 sets the RF DAC value.

For Type 8 (Non Quad U32) Special value1 sets the internal value to report.

For Type 9 (Non Quad F32) Special value1 sets the internal value to report.

All other Types do not use Special value1 at this time.

Special values for type 8 (Non Quad U32) defined to date:

0 = Time since power on in milliseconds

1 = Time since schedule started in milliseconds

2 = User Timer1 in milliseconds

3 = User Timer2 in milliseconds

10 = System Status

11 = Hardware Errors

12 = Hardware Warnings

20 = System Status new this scan

21 = Hardware Errors new this scan

22 = Hardware Warnings new this scan

40 = Input port Analog input 1 value in millivolts

41 = Input port Analog input 2 value in millivolts

42 = Input port Analog input 3 value in millivolts

43 = Input port Analog input 4 value in millivolts

44 = Input port Analog input 5 value in millivolts

45 = Input port Analog input 6 value in millivolts

46 = Input port Analog input 7 value in millivolts

47 = Input port Analog input 8 value in millivolts

48 = Input port Analog input 9 value in millivolts

50 = Digital Output State for all channels (bit0 is status relay state)

104 = Emission current in microamperes

107 = Anode potential in centivolts

108 = Electron energy in centivolts

109 = Focus potential in centivolts

111 = Filament potential in millivolts

112 = Filament current in millamperes

Special values for type 9 (Non Quad F32) defined to date:

0 = Ion Source total pressure (units as calibrated)

1 = Ion Source total pressure detector current in Amperes

2 = Temperature on CPU board in degrees Celsius

40 = Input port Analog input 1 value in volts

41 = Input port Analog input 2 value in volts

42 = Input port Analog input 3 value in volts

43 = Input port Analog input 4 value in volts
 44 = Input port Analog input 5 value in volts
 45 = Input port Analog input 6 value in volts
 46 = Input port Analog input 7 value in volts
 47 = Input port Analog input 8 value in volts
 48 = Input port Analog input 9 value in volts

type:	integer	\$message:	0x14
version:	0.5.0	\$type:	U16
		\$emul:	0
		\$index:	Channel
		\$default:	0
		\$flags:	PUBLIC
		\$range:	0:/electronicsInfo/massRange*100
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

223. Raw Channel # Stop Mass or special

[/mmsp/scanSetup/channel/<index#>/stopMassRaw](#)

Do Not Use This Command

Use StopMass instead. This StopMassRaw command is a low level command that is set automatically by StopMass and ChannelMode.

0-30000, special value2

This target queries or updates channel #'s starting mass or special value2.

The starting mass is in centiAMU. The special values are Channel Type dependent.

For Type 6 (cAMU User) Special value2 sets the DC equivalent mass value.

For Type 7 (DAC User) Special value2 sets the DC DAC value.

All other Types do not use Special value2 at this time.

type:	integer	\$message:	0x15
version:	0.5.0	\$type:	U16
		\$emul:	0
		\$index:	Channel
		\$default:	0
		\$flags:	PUBLIC
		\$range:	0:/electronicsInfo/massRange*100
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

224. Channel # points per AMU

</mmsp/scanSetup/channel/<index#>/ppamu>

Get or Set a channel's points per AMU. All values that divide into 100 with no remainder are allowed (1,2,4,5,10,20,25,50,100).

type:	integer	\$message:	0x16
version:	0.5.0	\$type:	U16
enum:	1,2,4,5,10,20,25,50,100	\$emul:	1
units:	ppu	\$index:	Channel
		\$default:	1
		\$flags:	PUBLIC
		\$range:	1,2,4,5,10,20,25,50,100
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

225. Channel # Dwell

</mmsp/scanSetup/channel/<index#>/dwell>

Get or Set a channel's dwell.

Dwell is the amount of time spent taking a reading durring a measurement.

The settable dwell range is from 1 to 16,384 milliseconds.

The time used making the measurement with the given dwell is specified as:

Dwell (ms)	Measurement Time (ms)
1	Dwell + 0.8
2	Dwell + 1.0
3	Dwell + 1.2
4-5	Dwell + 1.4
6-7	Dwell + 1.7
8-11	Dwell + 2.0
12-16384	Dwell + 3.2

Note: The most efficient use of available time is achieved by using one of the following dwells:

1,2,4,8,16,32,48,64,128,256,512,1024,2048,4096,8192,16384

type:	number	\$baseunits:	usec
version:	0.5.0	\$message:	0x17
divisibleBy:	0.001	\$type:	U32
units:	msec	\$emul:	1000
		\$index:	Channel
		\$default:	1000
		\$flags:	PUBLIC
		\$range:	/sensorFilter/minDwell:/sensorFilter/maxDwell
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

226. Channel # Local EM voltage

</mmsp/scanSetup/channel/<index#>/emVoltage>

This target queries or updates channel #'s local Electron Multiplier voltage setting.

This setting when non zero overrides the global Electron Multiplier voltage for all measurement(s) of this channel. Using this local setting will require a longer than normal initial measurement set-up time, thus lowering the number of measurement in a given time. When the channel is done the Electron Multiplier voltage will revert to the global value unless the next channel also has a local override.

type:	integer	\$message:	0x18
version:	2.0.0	\$type:	U16
units:	V	\$emul:	650
minimum:	650	\$index:	Channel
maximum:	3000	\$flags:	PUBLIC
		\$range:	650:3000
		\$length:	340
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

227. Channel # Local Focus voltage

</mmsp/scanSetup/channel/<index#>/focusVoltage>

This target queries or updates channel #'s local Focus voltage setting. This setting when non zero overrides the global Focus voltage for all measurement(s) of this channel. Using this local setting will require a longer than normal initial measurement set-up time, thus lowering the number of measurement in a given time. When the channel is done the Focus voltage will revert to the global value unless the next channel also has a local override.

type:	integer	\$message:	0x19
version:	2.0.0	\$type:	U16
notYetImplemented:	both	\$emul:	0
units:	V	\$index:	Channel
		\$flags:	PUBLIC
		\$range:	100
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

228. Channel # Local Ion Energy

</mmsp/scanSetup/channel/<index#>/ionEnergy>

This target queries or updates channel #'s local Ion Energy setting. This setting when non zero overrides the global Ion Energy for all measurement(s) of this channel. Using this local setting will require a longer than normal initial measurement set-up time, thus lowering the number of measurement in a given time. When the channel is done the Ion Energy will revert to the global value unless the next channel also has a local override.

type:	integer	\$message:	0x1C
version:	2.0.0	\$type:	U16
notYetImplemented:	both	\$emul:	5000
units:	mV	\$index:	Channel
minimum:	5	\$default:	5.000
maximum:	15	\$flags:	PUBLIC
		\$range:	5.000:15.000
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

229. Channel # Extra

</mmsp/scanSetup/channel/<index#>/extra>

Get or Set a channel's Extra count setting.

The extra readings are done for each measurement of the channel.

All ion source, filter and detector settings remain constant.

The extra measurements are performed at channel's set dwell.

Using a non-zero Extra value may be preferred over a single longer dwell if statistical analysis of the data is to be done. For very small signals the single

longer dwell will yield the best signal to noise ratio.
Extra points are reported individually in a scan.

type:	integer	\$message:	0x1D
version:	0.5.0	\$type:	U16
minimum:	0	\$emul:	0
maximum:	99	\$index:	Channel
		\$default:	0
		\$flags:	PUBLIC
		\$range:	0:99
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

230. Channel # Lead In

</mmsp/scanSetup/channel/<index#>/leadIn>

Get or Set a channel's Lead In measurement count. This is the number of Lead In readings before the first reportable measurement of the channel. All ion source, filter and detector settings remain constant. Lead In measurements are performed at the dwell specified in "LeadInDwell". Using a non-zero Lead In value may be useful for allowing extra settling time before usable measurements. The Lead In measured value is not reported in scans.

type:	integer	\$message:	0x1F
version:	0.5.0	\$type:	U16
minimum:	0	\$emul:	0
maximum:	99	\$index:	Channel
		\$default:	0
		\$flags:	PUBLIC
		\$range:	0:99
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

231. Channel # Measurement Enabled

</mmsp/scanSetup/channel/<index#>/enabled>

Enables a measurement.

If disabled the channel will be skipped during a scan. This may be useful to quickly change a scan without reprogramming the entire scan. To take effect, channels must be enabled AND fall within the range of ScanSetup/StartChannel and ScanSetup/StopChannel.

type:	string	\$message:	0x20
version:	0.5.0	\$type:	U16
enum:	False,True	\$index:	Channel
		\$default:	Enabled
		\$flags:	PUBLIC
		\$range:	False,True
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

232. Channel # Scale Factor

</mmsp/scanSetup/channel/<index#>/scaleFactor>

Get or Set a channel's Scale Factor setting.

This is a scalar multiplier of the channel's measurement(s).

type:	number	\$message:	0x22
version:	0.5.0	\$type:	F32
		\$emul:	1.0F
		\$index:	Channel
		\$default:	1
		\$flags:	PUBLIC
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

233. Channel # Equivalent Ion Current

</mmsp/scanSetup/channel/<index#>/equivIonFactor>

Get or Set a channel's Local Electron Multiplier Equivalent Ion Current Factor. When non zero the Electron Multiplier measurements are divided by this value. This function provides for a simpler transition from Faraday Cup to Electron Multiplier operation. When disabled actual currents are reported positive for Faraday Cup, negative for Electron Multiplier.

type:	number	\$message:	0x26
version:	0.5.0	\$type:	F32
		\$index:	Channel
		\$default:	Enabled
		\$flags:	PUBLIC
		\$range:	-1e15:1e15
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

234. Channel # Digital Output to Map

</mmsp/scanSetup/channel/<index#>/digOutNum>

Get or Set a given channel's mapped Digital Output number. Digital Output 0 is the native Relay control.

Other outputs may become available with option cards in the future.

type:	integer	\$message:	0x27
version:	2.0.0	\$type:	S16
minimum:	-1	\$index:	Channel
maximum:	17	\$flags:	PUBLIC
		\$range:	-1:17
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

235. Digital Output Upper Threshold

</mmsp/scanSetup/channel/<index#>/digOutThresUpper>

Get or Set a given channel's Digital Output Upper Threshold value.

type:	number	\$message:	0x28
version:	2.0.0	\$type:	F32
		\$index:	Channel
		\$flags:	PUBLIC
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

236. Digital Output Lower Threshold

</mmsp/scanSetup/channel/<index#>/digOutThresLower>

Get or Set a given channel's Digital Output Lower Threshold value.

type:	number	\$message:	0x29
version:	2.0.0	\$type:	F32
		\$index:	Channel
		\$flags:	PUBLIC
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

237. Channel # Start Mass

</mmsp/scanSetup/channel/<index#>/startMass>

Channel Start Mass

type: number \$baseunits: cAMU

version:	0.5.0	\$message:	0x32
divisibleBy:	0.01	\$type:	U16
units:	AMU	\$emul:	0
		\$index:	Channel
		\$default:	0
		\$flags:	SPECIAL PUBLIC
		\$range:	0:/electronicsInfo/massRange
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

238. Channel # Stop Mass

</mmsp/scanSetup/channel/<index#>/stopMass>

Channel Stop Mass

type:	number	\$baseunits:	cAMU
version:	0.5.0	\$message:	0x33
divisibleBy:	0.01	\$type:	U16
units:	AMU	\$emul:	0
		\$index:	Channel
		\$default:	0
		\$flags:	SPECIAL PUBLIC
		\$range:	0:/electronicsInfo/massRange
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

239. Channel Mode

</mmsp/scanSetup/channel/<index#>/channelMode>

Convenience message for setting special channels. This provides a cleaner way to set channelType, startMassRaw and stopMassRaw.

Setting the ChannelMode may have side effects, depending on what you set it to. You may change the

values of Dwell, Extra and Leadin after setting channelMode.

Possible values are:

Sweep	
Single	
Baseline	Sets Extra and Leadin to 0
TPQuad	
camuUser	
dacUser	
Time	
Timestamp	
UserTimer1	
UserTimer2	
SystemStatus	U32 bitfield
HardwareErrors	U32 bitfield
HardwareWarnings	U32 bitfield
SystemStatus2	U32 bitfield
NewSystemStatus	U32 bitfield
NewHardwareErrors	U32 bitfield
NewHardwareWarnings	U32 bitfield
AnalogInput1	U32 value in millivolts
AnalogInput2	U32 value in millivolts
AnalogInput3	U32 value in millivolts
AnalogInput4	U32 value in millivolts
AnalogInput5	U32 value in millivolts
AnalogInput6	U32 value in millivolts
AnalogInput7	U32 value in millivolts
AnalogInput8	U32 value in millivolts
AnalogInput9	U32 value in millivolts
AnalogInput0V	F32 value in volts
AnalogInput1V	F32 value in volts
AnalogInput2V	F32 value in volts
AnalogInput3V	F32 value in volts
AnalogInput4V	F32 value in volts
AnalogInput5V	F32 value in volts
AnalogInput6V	F32 value in volts
AnalogInput7V	F32 value in volts
AnalogInput8V	F32 value in volts
AnalogInput9V	F32 value in volts
AnalogOutput0V	F32 value in volts
AnalogOutput1V	F32 value in volts
AnalogOutput2V	F32 value in volts
AnalogOutput3V	F32 value in volts
AnalogOutput4V	F32 value in volts

AnalogOutput5V	F32 value in volts
AnalogOutput6V	F32 value in volts
AnalogOutput7V	F32 value in volts
ExternalGaugePressure	F32 value in pressure
ExternalGaugeVoltage	F32 value in voltage
DigitalOutput	U32 bitfield
DigitalInput	U32 bitfield
DigitalOutputNew	U32 bitfield
DecMSType	
DecState	
DecErrors	
DecWarnings	
EmissionCurrent	
DecEmFront	
DecEmBack	
AnodePotential	
ElectronEnergy	
FocusPotential	
DecSpare	
FilamentPotential	
FilamentCurrent	
Reserved1	
Reserved2	
Reserved3	
Reserved4	
Reserved5	
Reserved6	
Reserved7	
Reserved8	
Reserved9	
TotalPressure	Sets Extra and Leadin to 0, Dwell to 2 ms
TotalPressureDetectorCurrent	
CPUBoardTemperature	
Reserved10	
Reserved11	
ElectrometerPotential	
FixedNumber	
type:	string
version:	0.5.0
enum:	Sweep,Single,Baseline,TPQuad,camuUser,dacUser, Time,TimeStamp,UserTimer1,UserTimer2,SystemStatus, HardwareErrors,HardwareWarnings,NewSystemStatus, NewHardwareErrors,NewHardwareWarnings, AnalogInput1,AnalogInput2,AnalogInput3, AnalogInput4,AnalogInput5,AnalogInput6, AnalogInput7,AnalogInput8,AnalogInput9, AnalogInput1V,AnalogInput2V,AnalogInput3V,

	AnalogInput4V,AnalogInput5V,AnalogInput6V,AnalogInput7V,AnalogInput8V,AnalogInput9V,AnalogOutput0V,AnalogOutput1V,AnalogOutput2V,AnalogOutput3V,AnalogOutput4V,AnalogOutput5V,AnalogOutput6V,AnalogOutput7V,ExternalGaugePressure,ExternalGaugeVoltage,DigitalOutput,DigitalInput,DigitalOutputNew,decMstype,decState,decErrors,decWarnings,EmissionCurrent,decEmFront,decEmBack,AnodePotential,ElectronEnergy,FocusPotential,decSpare,FilamentPotential,FilamentCurrent,VSC_Status,VSC_Issues,VSC_TurboSpeed,TotalPressure,TotalPressureDetectorCurrent,CPUBoardTemperature,AuxIO_AnalogIn,VSC_ForelinePressure,VSC_ProcessPressure,ElectrometerPotential,FixedNumber
\$message:	0x34
\$type:	U16
\$emul:	1
\$index:	channel
\$flags:	SPECIAL PUBLIC
\$range:	Sweep,Single,Baseline,TPQuad,camuUser,dacUser,Time,Timestamp,UserTimer1,UserTimer2,SystemStatus,HardwareErrors,HardwareWarnings,NewSystemStatus,NewHardwareErrors,NewHardwareWarnings,AnalogInput1,AnalogInput2,AnalogInput3,AnalogInput4,AnalogInput5,AnalogInput6,AnalogInput7,AnalogInput8,AnalogInput9,AnalogInput1V,AnalogInput2V,AnalogInput3V,AnalogInput4V,AnalogInput5V,AnalogInput6V,AnalogInput7V,AnalogInput8V,AnalogInput9V,AnalogOutput0V,AnalogOutput1V,AnalogOutput2V,AnalogOutput3V,AnalogOutput4V,AnalogOutput5V,AnalogOutput6V,AnalogOutput7V,ExternalGaugePressure,ExternalGaugeVoltage,DigitalOutput,DigitalInput,DigitalOutputNew,decMstype,decState,decErrors,decWarnings,EmissionCurrent,decEmFront,decEmBack,AnodePotential,ElectronEnergy,FocusPotential,decSpare,FilamentPotential,FilamentCurrent,VSC_Status,VSC_Issues,VSC_TurboSpeed,TotalPressure,TotalPressureDetectorCurrent,CPUBoardTemperature,AuxIO_AnalogIn,VSC_ForelinePressure,VSC_ProcessPressure,ElectrometerPotential,FixedNumber
\$readmode:	locked
\$writemode:	control
\$execmode:	null

240. Channel # Analog Output to Map

[/mmssp/scanSetup/channel/<index#>/aONum](#)

Get or Set which analog output is mapped to the channels measured value. When not mapping a channel's measured value to an analog output set this to -1.

Note: Analog outputs are only available when an optional extended IO card is installed.

See also: Figure 7 Analog Output Block Diagram

type:	integer	\$message:	0x50
version:	1.4.2	\$type:	S16
		\$index:	Channel
		\$flags:	PUBLIC
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

241. Analog Output Input Low Limit

[/mmssp/scanSetup/channel/<index#>/aOInputLowLimit](#)

Get or Set a given channel's "Analog Output" Scaling Input Low Limit. This is the channel's measured value associated with an analog output value of zero volts. When the mapped channel's measured value is less than or equal to the Low Limit the analog output will be the greater of the "Floor" or zero volts.

See also: Figure 7 Analog Output Block Diagram

type:	number	\$message:	0x51
version:	1.4.2	\$type:	F32
		\$index:	Channel
		\$flags:	PUBLIC
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

242. Analog Output Input High Limit

[/mmssp/scanSetup/channel/<index#>/aOInputHighLimit](#)

Get or Set a given channel's "Analog Output" Scaling Input High Limit. This is the channel's measured value associated with an analog output value of ten volts. When the mapped channel's measured value is greater than or equal to the High Limit the analog output will be the lesser value of the "Ceiling" or ten volts.

See also: Figure 7 Analog Output Block Diagram

type:	number	\$message:	0x52
version:	1.4.2	\$type:	F32
		\$index:	Channel
		\$flags:	PUBLIC
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

243. Analog Output Mode

[/mmsp/scanSetup/channel/<index#>/aOMode](#)

Get or Set a given channel's "Analog Output" Mode.

Available Modes: Floor, Ceiling, Zero, Direct, Log10 and Linear

0 - Floor mode: Returns the floor value.

1 - Ceiling mode: Returns the ceiling value.

2 - Zero mode: Returns a 0 value.

3 - Direct: Sets the scaled out value equal to the scaled in value

4 - Log10: Sets the scaled out value equal to $C1 + C2 * \text{Log10}(\text{Scale In})$

5 - Linear: Sets the scaled out value equal to $\text{Slope} * (\text{Scale In} - \text{Intercept})$

Note:

The Log10 mode output is controlled by the following expression:

$\text{Scale Out} = C1 + C2 * \text{Log10}(\text{Scale In})$

Where:

Scale In = Set from channel measurement or manually by target

Scale Out Max Value is 10.0 V

Scale Out Min Value is 0.0 V

Scale In High Limit = user picked value in Scale In range greater than Scale In Low Limit

Scale In Low Limit = user picked value in Scale In range less than Scale Out High Limit

C1 and C2 are calculated automatically when a scan is started with a channel using

analog outputs in Log10 mode. The following is the math used:

$C2 = \text{Scale Out Max Value} / (\text{Log10}(\text{Scale In High Limit}) - \text{Log10}(\text{Scale In Low Limit}))$

$C1 = -C2 * \text{Log10}(\text{Scale In Low Limit})$

Note:

The Linear mode output is controlled by the following expression:

$\text{Scale Out} = \text{Slope} * (\text{Scale In} - \text{Intercept})$

Where:

Scale In = Set from channel measurement or manually by target

Scale Out Max Value is 10.0 V

Scale Out Min Value is 0.0 V

Scale In High Limit = user picked value in Scale In range greater than Scale In Low Limit

Scale In Low Limit = user picked value in Scale In range less than Scale Out High Limit

Slope and Intercept are calculated automatically when a scan is started with a channel using

analog outputs in Linear mode. The following is the math used:

$\text{Slope} = \text{Scale Out Max Value} / ((\text{Scale In High Limit}) - (\text{Scale In Low Limit}))$

$\text{Intercept} = (\text{Scale In Low Limit})$

Note: Set channel's "Analog Output" Mode to 3 (Direct) if using the [/mmsp/analogOutput/](#) targets to set

scaling

See also: Figure 7 Analog Output Block Diagram

type:	integer	\$message:	0x53
version:	1.4.2	\$type:	U8
		\$index:	Channel
		\$flags:	PUBLIC
		\$length:	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

244. Analog Input Parameters

[/mmssp/analogInput](#)

type: object \$block: 0x20XX

245. Analog Input Channels

[/mmssp/analogInput/ioChannelCount](#)

The MMSP CPU single differential analog input is located on pins 9 and 10 of the "Aux I/O" connector.

Pin Name

9 Ain(+)

10 Ain(-)

The JSON target for Analog-In1 is

<http://Your MMSP IP/mmssp/analogInput/ioChannel/1/scaledValue/get>

Various scaling parameters can be set using the API and JSON.

For example, referring to Figure 6, the following JSON target will scale the raw value by 2.0

<http://Your MMSP IP/mmssp/analogInput/ioChannel/1/designMultiplier/set?2.0>

<http://Your MMSP IP/mmssp/analogInput/ioChannel/1/scaledvalue/get>

type:	integer	\$message:	0x10
version:	0.5.0	\$type:	U32
readonly:	true	\$emul:	32
		\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

246. Analog Input Default Load

[/mmssp/analogInput/defaultLoad](#)

Load and activate the genus specific analog input non volatile parameters.

type:	integer	\$message:	0x18
version:	1.4.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

247. Analog Input Events Default Load

[/mmssp/analogInput/eventsDefaultLoad](#)

Load and activate the default analog input event settings.

type:	integer	\$message:	0x19
version:	1.4.2	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

248. Analog Input Channels Available

[/mmssp/analogInput/channelsAvailable](#)

Get a bit representation of the analog input channels available to the user. The bit definitions are tabulated below. A returned bit of 1 means the channel is available.

Bit(s)	Description
31:6	0/1 = Reserved (Not Available)
5	Analog Input 5 "Extended IO" pin 55(+), pin 56(-)
4	Analog Input 4 "Extended IO" pin 53(+), pin 54(-)
3	Analog Input 3 "Extended IO" pin 51(+), pin 52(-)
2	Analog Input 2 "Extended IO" pin 49(+), pin 50(-)
1	Analog Input 1 "Aux IO" pin 9(+), pin 10(-)
0	Reserved - Not Available

type:	integer	\$message:	0x1A
version:	1.4.2	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

249. Event Count

[/mmssp/analogInput/eventCount](#)

Number of analog input events available.

type:	integer	\$message:	0x40
version:	0.17.0	\$type:	U16
readonly:	true	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

250. aiEvent

[/mmssp/analogInput/aiEvent](#)

type:	array	\$range:	0:9
minIndex:	0		
minItems:	0		
maxItems:	32		

Array of objects with following properties:

type: object

251. Event to AI mapping

[/mmssp/analogInput/aiEvent/<index#>/eventMapping](#)

Get or Set the analog input channel mapped to the event. Set to -1 to disable.

type:	integer	\$message:	0x41
version:	0.17.0	\$type:	S16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

252. Event Threshold 1

</mmsp/analogInput/aiEvent/<index#>/eventThreshold1>

Get or Set the threshold for the mapped analog input comparison.

type:	number	\$message:	0x42
version:	0.17.0	\$type:	F32
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

253. Event Mode 1

</mmsp/analogInput/aiEvent/<index#>/eventMode1>

Get or Set the mode used to compare analog input value to threshold.

0 = none

1 = input greater than threshold

2 = input less than threshold

3 = input transitioned from less than to greater than threshold

4 = input transitioned from greater than to less than threshold.

type:	integer	\$message:	0x43
version:	0.17.0	\$type:	U16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

254. Event Action 1

</mmsp/analogInput/aiEvent/<index#>/eventAction1>

Get or Set the action to take when the mode specified threshold comparison is true.

0 = none

1 = Emission and EM off

2 = EM off

3 = Emission on

4 = Emission and EM on

5 = EM on

type:	integer	\$message:	0x44
version:	0.17.0	\$type:	U16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

255. Event Action Count 1

[/mmssp/analogInput/aiEvent/<index#>/eventActionCount1](#)

Get or Set the count of actions taken.

type:	integer	\$message:	0x45
version:	0.17.0	\$type:	U32
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

256. Event Threshold 2

[/mmssp/analogInput/aiEvent/<index#>/eventThreshold2](#)

Get or Set the threshold for the mapped analog input comparison.

type:	number	\$message:	0x46
version:	0.17.0	\$type:	F32
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

257. Event Mode 2

</mmsp/analogInput/aiEvent/<index#>/eventMode2>

Get or Set the mode used to compare analog input value to threshold.

0 = none

1 = input greater than threshold

2 = input less than threshold

3 = input transitioned from less than to greater than threshold

4 = input transitioned from greater than to less than threshold.

type:	integer	\$message:	0x47
version:	0.17.0	\$type:	U16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

258. Event Action 2

</mmsp/analogInput/aiEvent/<index#>/eventAction2>

Get or Set the action to take when the mode specified threshold comparison is true.

0 = none

1 = Emission and EM off

2 = EM off

3 = Emission on

4 = Emission and EM on

5 = EM on

type:	integer	\$message:	0x48
version:	0.17.0	\$type:	U16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

259. Event Action Count 2

</mmssp/analogInput/aiEvent/<index#>/eventActionCount2>

Get or Set the count of actions taken.

type:	integer	\$message:	0x49
version:	0.17.0	\$type:	U32
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

260. Event Threshold 3

</mmssp/analogInput/aiEvent/<index#>/eventThreshold3>

Get or Set the threshold for the mapped analog input comparison.

type:	number	\$message:	0x4A
version:	0.17.0	\$type:	F32
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

261. Event Mode 3

</mmssp/analogInput/aiEvent/<index#>/eventMode3>

Get or Set the mode used to compare analog input value to threshold.

0 = none

1 = input greater than threshold

2 = input less than threshold

3 = input transitioned from less than to greater than threshold

4 = input transitioned from greater than to less than threshold.

type:	integer	\$message:	0x4B
version:	0.17.0	\$type:	U16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

262. Event Action 3

[/mmssp/analogInput/aiEvent/<index#>/eventAction3](#)

Get or Set the action to take when the mode specified threshold comparison is true.

0 = none

1 = Emission and EM off

2 = EM off

3 = Emission on

4 = Emission and EM on

5 = EM on

type:	integer	\$message:	0x4C
version:	0.17.0	\$type:	U16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

263. Event Action Count 3

[/mmssp/analogInput/aiEvent/<index#>/eventActionCount3](#)

Get or Set the count of actions taken.

type:	integer	\$message:	0x4D
version:	0.17.0	\$type:	U32
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

264. Event Threshold 4

</mmsp/analogInput/aiEvent/<index#>/eventThreshold4>

Get or Set the threshold for the mapped analog input comparison.

type:	number	\$message:	0x4E
version:	0.17.0	\$type:	F32
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

265. Event Mode 4

</mmsp/analogInput/aiEvent/<index#>/eventMode4>

Get or Set the mode used to compare analog input value to threshold.

0 = none

1 = input greater than threshold

2 = input less than threshold

3 = input transitioned from less than to greater than threshold

4 = input transitioned from greater than to less than threshold.

type:	integer	\$message:	0x4F
version:	0.17.0	\$type:	U16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

266. Event Action 4

</mmsp/analogInput/aiEvent/<index#>/eventAction4>

Get or Set the action to take when the mode specified threshold comparison is true.

0 = none

1 = Emission and EM off

2 = EM off

3 = Emission on

4 = Emission and EM on

5 = EM on

type:	integer	\$message:	0x50
version:	0.17.0	\$type:	U16
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

267. Event Action Count 4

[/mmsp/analogInput/aiEvent/<index#>/eventActionCount4](#)

Get or Set the count of actions taken.

type:	integer	\$message:	0x51
version:	0.17.0	\$type:	U32
		\$index:	aiEvent
		\$flags:	NVM PUBLIC
		\$length:	0:9
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

268. ioChannel

[/mmsp/analogInput/ioChannel](#)

type:	array	\$range:	0:!IOChannelCount
minIndex:	0		
minItems:	0		
maxItems:	32		

Array of objects with following properties:

type: object

269. Analog Input y Raw value

[/mmsp/analogInput/ioChannel/<index#>/rawValue](#)

RawValue from ADC in millivolts for Chn#1 and volts for Chn#2 – Chn#9.

type:	number	\$message:	0x11
version:	0.5.0	\$type:	F32
readonly:	true	\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

270. Analog Input y Design Offset

[/mmsp/analogInput/ioChannel/<index#>/designOffset](#)

Value added to the RawValue as a design offset.

type:	number	\$message:	0x12
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$default:	0
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

AnalogIn[1]: 0 to 5000 mV => DesignMultiplier: 0.002 => AppValue: 0-10 V

271. Analog Input y Error Offset

</mmssp/analogInput/ioChannel/<index#>/errorOffset>

Value added to the RawValue as a error correction offset.

type:	number	\$message:	0x13
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$default:	0
		\$flags:	NVM PUBLIC
		\$length:	0:!!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

272. Analog Input y Design Multiplier

</mmssp/analogInput/ioChannel/<index#>/designMultiplier>

Value RawValue multiplied by as a design scaling factor.

In1: Standard = 0.002

In2-9: Standard = 1

In1 is on a different Board as In2-9

AnalogIn[1]: 0 to 5000 mV => DesignMultiplier: 0.002 => AppValue: 0-10 V

type:	number	\$message:	0x14
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$default:	1
		\$flags:	NVM PUBLIC
		\$length:	0:!!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

273. Analog Input y Error Multiplier

[/mmsp/analogInput/ioChannel/<index#>/errorMultiplier](#)

Value RawValue multiplied by as a gain error correction.

type:	number	\$message:	0x15
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$default:	1
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

274. Analog Input y Corrected value

[/mmsp/analogInput/ioChannel/<index#>/correctedValue](#)

The analog in value after Design Offset, Error Offset, Design Multiplier and Error Multiplier have been applied.

type:	number	\$message:	0x16
version:	0.5.0	\$type:	F32
readonly:	true	\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

275. Analog Input y Mode

[/mmsp/analogInput/ioChannel/<index#>/mode](#)

Get or Set the Analog Input scaling mode

Available Modes: Floor, Ceiling, Zero or Direct

Floor mode: Returns the floor value.

Ceiling mode: Returns the ceiling value.

Zero mode: Returns a 0 value.

Direct: Sets the scaled out value equal to the scaled in value

type:	string	\$message:	0x20
version:	0.5.0	\$type:	U8
enum:	Floor,Ceiling, Zero,PassThrough, Exponential, Scaling_2,Scaling_3	\$index:	ioChannel
		\$default:	PassThrough
		\$flags:	NVM PUBLIC
		\$range:	Floor,Ceiling, Zero,PassThrough, Exponential, Scaling_2,Scaling_3
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

276. Analog Input y Scale message count

[/mmsp/analogInput/ioChannel/<index#>/scaleMsgCount](#)

type:	integer	\$message:	0x21
version:	0.5.0	\$type:	U16
notYetImplemented:	both	\$emul:	0
minimum:	0	\$index:	ioChannel
maximum:	32	\$flags:	PUBLIC
		\$range:	0:32
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

277. Analog Input y Scale message Id(s)

[/mmsp/analogInput/ioChannel/<index#>/scaleMsgId](#)

type:	integer	\$message:	0x22
version:	0.5.0	\$type:	U32
notYetImplemented:	both	\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0: !IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

278. Analog Input y Scaled value

[/mmsp/analogInput/ioChannel/<index#>/scaledValue](#)

The value after mode scaling complete.

Refer to Figure 6.

type:	number	\$message:	0x23
version:	0.5.0	\$type:	F32
readonly:	true	\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0: !IOChannelCount
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

279. Analog Input y Bypass

[/mmsp/analogInput/ioChannel/<index#>/bypass](#)

If Bypass is 0, the analog in ScaledValue will be passed on to the Analog Input Application.

If Bypass is 1, the value contained in AppValue will be passed on to the Analog Input Application.

type:	string	\$message:	0x24
version:	0.5.0	\$type:	U8
enum:	NoBypass,Bypass	\$index:	ioChannel
		\$default:	NoBypass
		\$flags:	NVM PUBLIC
		\$range:	NoBypass,Bypass
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

280. Analog Input y Application Value

</mmsp/analogInput/ioChannel/<index#>/appValue>

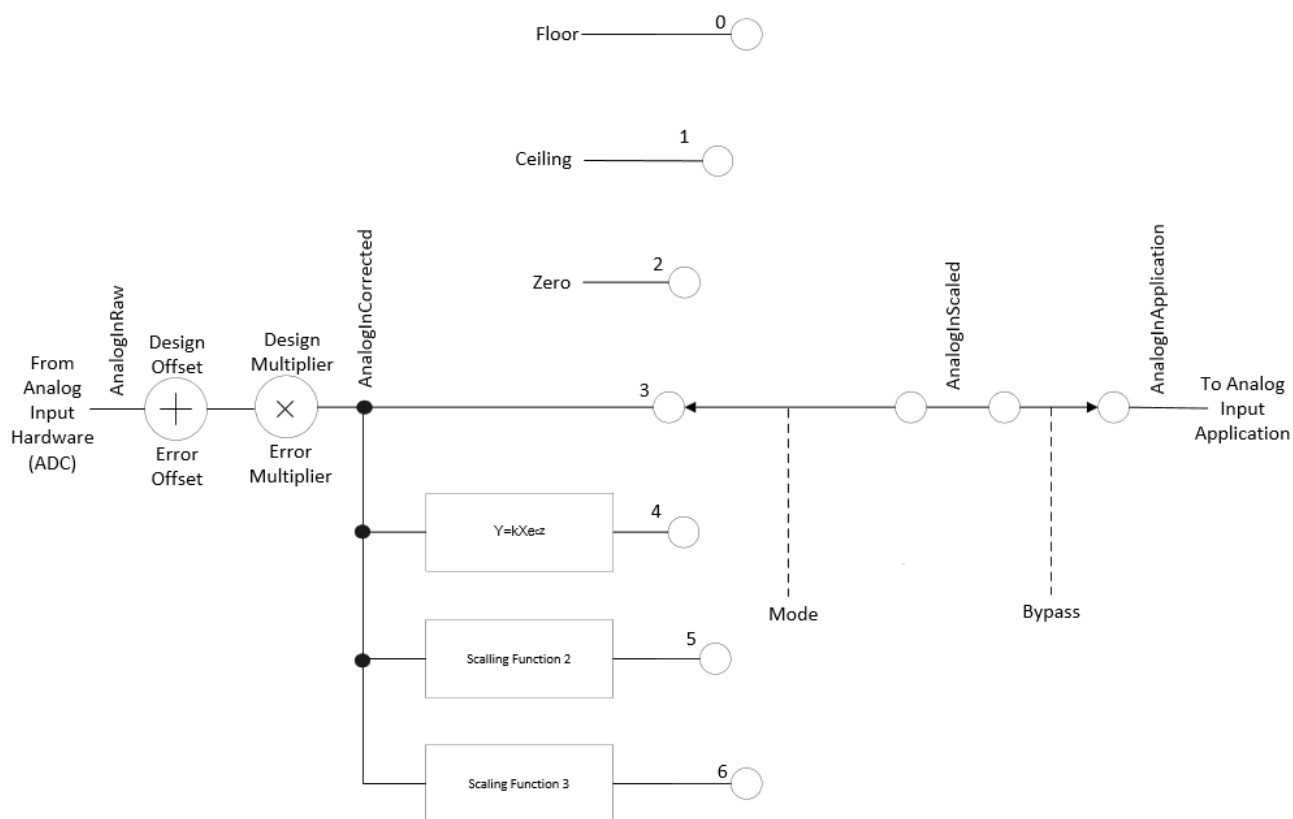


Figure 6 Analog Input Block Diagram

type:	number	\$message:	0x25
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

281. Analog Output Parameters

[/mmssp/analogOutput](#)

type: object \$block: 0x21XX

282. Analog Output Channels

[/mmssp/analogOutput/ioChannelCount](#)

Refer to Figure 7

Note: in the following **y** represents the channel number.

Analog Output Channels

Analog Output y Raw

Analog Output y Bypass

'NoBypass'

'Bypass'

Analog Output y Scale In Value

Analog Output y Ceiling

Analog Output y Floor

Analog Output y Scale Mode

'Floor'

'Ceiling'

'Zero'

'Direct'

'Log10'

Analog Out y Scale Out Value

Analog Output y Converter Gain, default is 1

Analog Output y Converter Offset, default is 0

Analog Output y Hardware Value

type:	integer	\$message:	0x10
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	10
		\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

283. Analog Outputs Default Load

[/mmssp/analogOutput/defaultLoad](#)

Load and activate the genus specific analog output non volatile parameters.

type:	integer	\$message:	0x18
version:	1.4.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

284. Analog Output Channels Available

[/mmsp/analogOutput/channelsAvailable](#)

Get a bit representation of the analog output channels available to the user.

A returned bit of 1 means the channel is available.

The bit definitions are tabulated below.

Bit(s)	Description		
31:4	Reserved (Not Available)		
3	Analog Output 3		
2	Analog Output 2		
1	Analog Output 1		
0	Analog Output 0		

type:	integer	\$message:	0x1A
version:	1.4.2	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

285. ioChannel

[/mmsp/analogOutput/ioChannel](#)

type:	array	\$range:	0:!IOChannelCount
minIndex:	0		
minItems:	0		
maxItems:	32		

Array of objects with following properties:

type: object

286. Analog Output # Raw

</mmsp/analogOutput/ioChannel/<index#>/raw>

The value presented to the Analog Out function. Data originates from the Analog Output Application. The Analog Out function processes the data by scaling, adding offsets, or fitting to user supplied algorithms.

type:	number	\$message:	0x11
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

287. Analog Output # Bypass

</mmsp/analogOutput/ioChannel/<index#>/bypass>

If Bypass is 0, the Analog Out Raw value will be processed and passed on to the Analog Out circuitry.

If Bypass is 1, the value contained in ScaleOutValue will be processed and passed on to the Analog Out circuitry.

type:	string	\$message:	0x12
version:	0.5.0	\$type:	U8
enum:	NoBypass,Bypass	\$index:	ioChannel
		\$default:	NoBypass
		\$flags:	NVM PUBLIC
		\$range:	NoBypass,Bypass
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

288. Analog Output # Scale In Value

</mmsp/analogOutput/ioChannel/<index#>/scaleInValue>

A value that can be set and used in place of the Analog Out Raw value.

type:	number	\$message:	0x13
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

289. Analog Output # Scale Mode

</mmsp/analogOutput/ioChannel/<index#>/scaleMode>

Get or Set a given channel's "Analog Output" Mode.

Available Modes: Floor, Ceiling, Zero, Direct, Log10 and Linear

0 - Floor mode: Returns the floor value.

1 - Ceiling mode: Returns the ceiling value.

2 - Zero mode: Returns a 0 value.

3 - Direct: Sets the scaled out value equal to the scaled in value

4 - Log10: Sets the scaled out value equal to $C1 + C2 * \text{Log10}(\text{Scale In})$

5 - Linear: Sets the scaled out value equal to $\text{Slope} * (\text{Scale In} - \text{Intercept})$

Note:

The Log10 mode output is controlled by the following expression:

$\text{Scale Out} = C1 + C2 * \text{Log10}(\text{Scale In})$

Where:

Scale In = Set from channel measurement or manually by target

Scale Out Max Value is 10.0 V

Scale Out Min Value is 0.0 V

Scale In High Limit = user picked value in Scale In range greater than Scale In Low Limit

Scale In Low Limit = user picked value in Scale In range less than Scale Out High Limit

C1 and C2 are calculated automatically when a scan is started with a channel using

analog outputs in Log10 mode. The following is the math used:

$C2 = \text{Scale Out Max Value} / (\text{Log10}(\text{Scale In High Limit}) - \text{Log10}(\text{Scale In Low Limit}))$

$C1 = -C2 * \text{Log10}(\text{Scale In Low Limit})$

Note:

The Linear mode output is controlled by the following expression:

$\text{Scale Out} = \text{Slope} * (\text{Scale In} - \text{Intercept})$

Where:

Scale In = Set from channel measurement or manually by target

Scale Out Max Value is 10.0 V

Scale Out Min Value is 0.0 V

Scale In High Limit = user picked value in Scale In range greater than Scale In Low Limit

Scale In Low Limit = user picked value in Scale In range less than Scale Out High Limit

Slope and Intercept are calculated automatically when a scan is started with a channel using

analog outputs in Linear mode. The following is the math used:

Slope = Scale Out Max Value/((Scale In High Limit) - (Scale In Low Limit))

Intercept = (Scale In Low Limit)

Note: Set channel's "ScaleMode" to 3 (Direct) if using the /mmsp/scanSetup/ targets to set scaling

type:	string	\$message:	0x20
version:	0.5.0	\$type:	U8
enum:	Floor,Ceiling,Zero,Direct,Log10,Linear	\$index:	ioChannel
		\$default:	Direct
		\$flags:	NVM PUBLIC
		\$range:	Floor,Ceiling,Zero,Direct,Log10,Linear
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

290. Analog Output # Ceiling {Max}

[/mmsp/analogOutput/ioChannel/<index#>/ceiling](#)

Analog Output y Scale Mode 'Ceiling'.

Highest scaled Analog Output value.

Value is in volts.

type:	number	\$message:	0x21
version:	0.5.0	\$type:	F32
units:	V	\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

291. Analog Output # Floor {Min}

</mmsp/analogOutput/ioChannel/<index#>/floor>

Analog Output y Scale Mode 'Floor'.

Lowest scaled Analog Output value.

Value is in volts.

type:	number	\$message:	0x22
version:	0.5.0	\$type:	F32
units:	V	\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

292. Analog Out y Scale Out Value

</mmsp/analogOutput/ioChannel/<index#>/scaleOutValue>

The value after mode scaling complete.

Refer to Figure 7.

type:	number	\$message:	0x2C
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

293. Analog Output # Design Offset

</mmsp/analogOutput/ioChannel/<index#>/designOffset>

Value added to the mode scaled output as a design offset.

type:	number	\$message:	0x30
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$default:	0
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

294. Analog Output # Error Offset

</mmsp/analogOutput/ioChannel/<index#>/errorOffset>

Value added to the mode scaled output as an error correction

type:	number	\$message:	0x31
version:	1.4.0	\$type:	F32
		\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

295. Analog Output # Design Gain

</mmsp/analogOutput/ioChannel/<index#>/designGain>

Multiplier of the mode scaled output used as a designed gain.

type:	number	\$message:	0x32
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$default:	1
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

296. Analog Output # Error Gain

</mmsp/analogOutput/ioChannel/<index#>/errorGain>

Multiplier of the mode scaled output used for gain error correction.

type:	number	\$message:	0x33
version:	1.4.0	\$type:	F32
		\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0: !IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

297. Analog Output # Hardware Value

</mmsp/analogOutput/ioChannel/<index#>/hardwareValue>

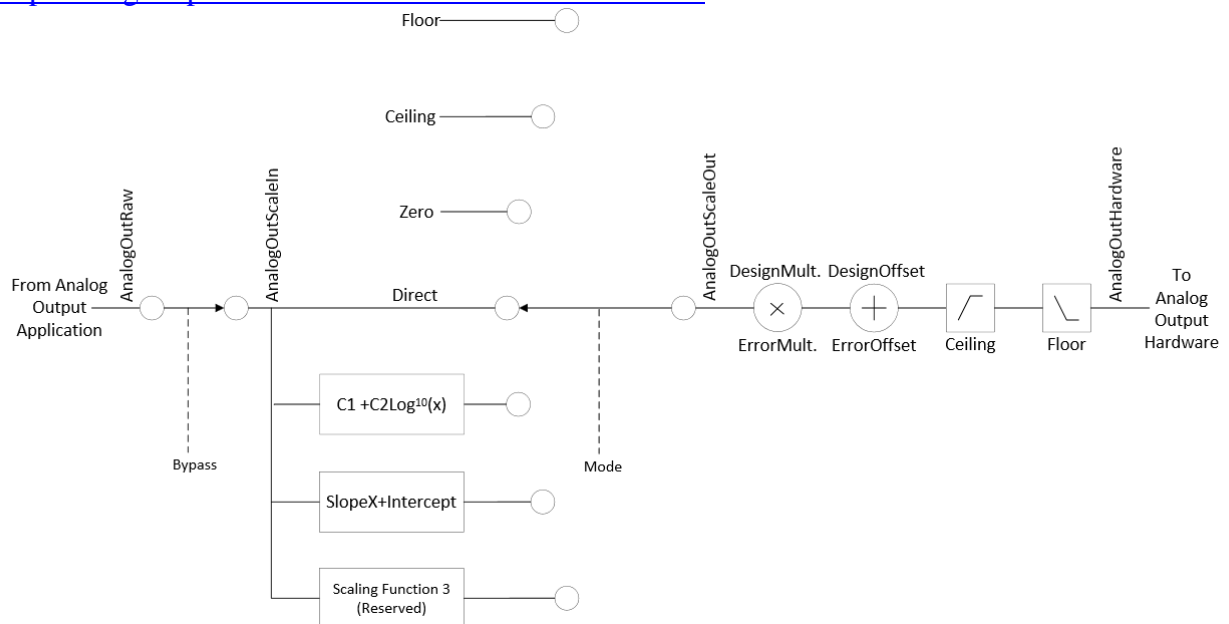


Figure 7 Analog Output Block Diagram

Post processed value that is passed to Analog Out circuitry.

type:	number	\$message:	0x34
version:	0.5.0	\$type:	F32
		\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0: !IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

298. Analog Output # Input Low Limit

</mmsp/analogOutput/ioChannel/<index#>/inputLowLimit>

Get or Set a given channel's "Analog Output" Scaling Input Low Limit. This is the channel's measured value associated with an analog output value of zero volts. When the mapped channel's measured value is less than or equal to the Low Limit the analog output will be the greater of the "Floor" or zero volts.

See also: Figure 7 Analog Output Block Diagram

type:	number	\$message:	0x38
version:	1.4.2	\$type:	F32
		\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

299. Analog Output # Input High Limit

</mmsp/analogOutput/ioChannel/<index#>/inputHighLimit>

Get or Set a given channel's "Analog Output" Scaling Input High Limit. This is the channel's measured value associated with an analog output value of ten volts. When the mapped channel's measured value is greater than or equal to the High Limit the analog output will be the lesser value of the "Ceiling" or ten volts.

See also: Figure 7 Analog Output Block Diagram

type:	number	\$message:	0x39
version:	1.4.2	\$type:	F32
		\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

300. Digital Input Parameters

</mmsp/digitalInput>

type: object \$block: 0x22XX

301. Digital Input Channels

</mmsp/digitalInput/ioChannelCount>

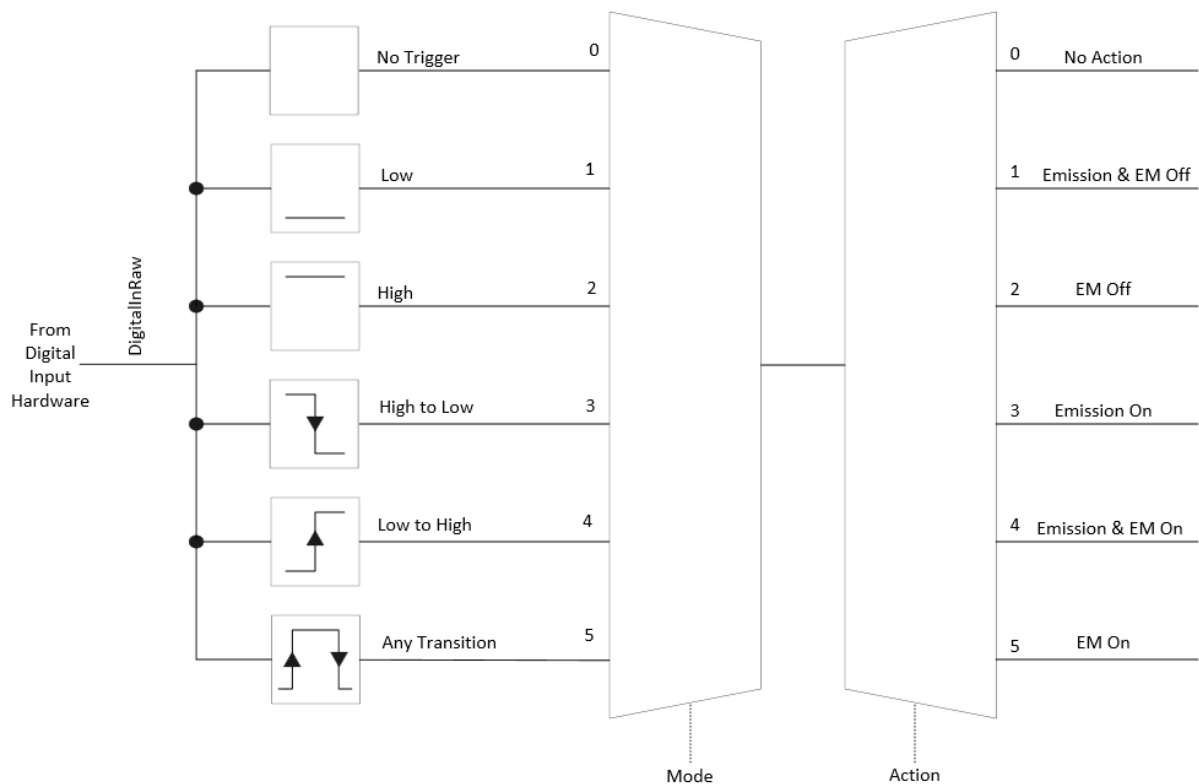


Figure 8 Digital Input Block Diagram

Digital Input Channels

Digital Input [ioChannel] Raw value

Digital Input [ioChannel] Mode

'NoTrigger'

'Low' voltage at input connection

'High' voltage at input connection

'HighToLow'

'LowToHigh'

'AnyTransition'

Digital Input [ioChannel] Action

'NoAction'

'EmissionAndEM_Off'

'EM_Off'

'EmissionOn'

'EmissionAndEM_On'

'EM_On'

type:	integer	\$message:	0x10
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	10
		\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

302. Extended IO Bit Config

[/mmsp/digitalInput/bitConfig](#)

Get or Set the Extended DIO bits configuration. If a bit is set to zero the corresponding DIO will be an output.

If a bit is set to one the corresponding DIO will be an input. Bits 0 and 1 are not configurable.

Bit(s)	Description
31:14	Reserved (Not Available)
13	Digital Input 13
12	Digital Input 12
11	Digital Input 11
10	Digital Input 10
9	Digital Input 9
8	Digital Input 8
7	Digital Input 7
6	Digital Input 6
5	Digital Input 5
4	Digital Input 4
3	Digital Input 3
2	Digital Input 2
1	Digital Input 1
0	Digital Input 0

type:	integer	\$message:	0x12
version:	1.4.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

303. Raw Value Bitwise

[/mmssp/digitalInput/rawValueBitwise](#)

Get the present value of all digital inputs in a bit per channel format.
A bit value of 1 indicates a high value.

Bit(s)	Description
31:14	Reserved (Not Available)
13	Digital Input 13
12	Digital Input 12
11	Digital Input 11
10	Digital Input 10
9	Digital Input 9
8	Digital Input 8
7	Digital Input 7
6	Digital Input 6
5	Digital Input 5
4	Digital Input 4
3	Digital Input 3
2	Digital Input 2
1	Digital Input 1
0	Digital Input 0

type:	integer	\$message:	0x13
version:	0.47.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

304. Digital Input Default Load

[/mmssp/digitalInput/defaultLoad](#)

Load and activate the genus specific digital input non volatile parameters.

type:	integer	\$message:	0x18
version:	1.1.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

305. Digital Input Channels Available

</mmsp/digitalInput/channelsAvailable>

Get a bit representation of the digital input channels available to the user.

The bit definitions are tabulated below. A returned bit value of 1 means the channel is available.

Bit(s)	Description
31:14	Reserved (Not Available)
13	Digital Input 13
12	Digital Input 12
11	Digital Input 11
10	Digital Input 10
9	Digital Input 9
8	Digital Input 8
7	Digital Input 7
6	Digital Input 6
5	Digital Input 5
4	Digital Input 4
3	Digital Input 3
2	Digital Input 2
1	Digital Input 1
0	Digital Input 0

type:	integer	\$message:	0x1A
version:	1.4.2	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

306. Digital Input Hardware Values

</mmsp/digitalInput/hardwareValues>

Get all Digital Input Hardware Values at once.

Each bit represents a Digital Input value.

The asserted state, represented by a bit value of 1, indicates the digital input pin is pulled to ground.

Bit(s)	Description
31:14	Reserved (Not Available)
13	Digital Input 13
12	Digital Input 12
11	Digital Input 11

10	Digital Input 10
9	Digital Input 9
8	Digital Input 8
7	Digital Input 7
6	Digital Input 6
5	Digital Input 5
4	Digital Input 4
3	Digital Input 3
2	Digital Input 2
1	Digital Input 1
0	Digital Input 0

type:	integer	\$message:	0x40
version:	1.4.0	\$type:	U16
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

307. ioChannel

[/mmsp/digitalInput/ioChannel](#)

type:	array	\$range:	0:!IOChannelCount
minIndex:	0		
minItems:	0		
maxItems:	32		

Array of objects with following properties:

type: object

308. Digital Input # Raw value

[/mmsp/digitalInput/ioChannel/<index#>/rawValue](#)

See Figure 8 Digital Input Block Diagram

Get Digital Input [ioChannel] value.

type:	integer	\$message:	0x11
version:	1.1.0	\$type:	U16
readonly:	true	\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

309. Trigger 1 Mode

[/mmsp/digitalInput/ioChannel/<index#>/trigger1Mode](#)

See Figure 8 Digital Input Block Diagram

Get or Set Digital Input [ioChannel] Trigger 1 Mode

0 = Disabled

1 = Low

2 = High

3 = Negative edge (HighToLow)

4 = Positive edge (LowToHigh)

5 = Any edge (HighToLow or LowToHigh)

type:	integer	\$message:	0x20
version:	1.1.0	\$type:	U8
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

310. Trigger 1 Action

</mmsp/digitalInput/ioChannel/<index#>/trigger1Action>

See Figure 8 Digital Input Block Diagram

Get or Set Digital Input [ioChannel] Trigger 1 Action

0 = No action

1 = Turn Emission and EM off

2 = Turn EM off

3 = Turn Emission on

4 = Turn Emission and EM on

5 = Turn EM on

type:	integer	\$message:	0x22
version:	1.1.0	\$type:	U8
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

311. Trigger 1 Action Count

</mmsp/digitalInput/ioChannel/<index#>/trigger1ActCnt>

Get or Set Trigger 1 Action Count

type:	integer	\$message:	0x23
version:	1.1.0	\$type:	U32
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

312. Trigger 2 Mode

</mmsp/digitalInput/ioChannel/<index#>/trigger2Mode>

See Figure 8 Digital Input Block Diagram

Get or Set Digital Input [ioChannel] Trigger 2 Mode

0 = Disabled

1 = Low

2 = High

3 = Negative edge (HighToLow)

4 = Positive edge (LowToHigh)

5 = Any edge (HighToLow or LowToHigh)

type:	integer	\$message:	0x24
version:	1.1.0	\$type:	U8
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

313. Trigger 2 Action

</mmsp/digitalInput/ioChannel/<index#>/trigger2Action>

See Figure 8 Digital Input Block Diagram

Get or Set Digital Input [ioChannel] Trigger 2 Action

0 = No action

1 = Turn Emission and EM off

2 = Turn EM off

3 = Turn Emission on

4 = Turn Emission and EM on

5 = Turn EM on

type:	integer	\$message:	0x26
version:	1.1.0	\$type:	U8
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

314. Trigger 2 Action Count
</mmsp/digitalInput/ioChannel/<index#>/trigger2ActCnt>

Get or Set Trigger 2 Action Count

type:	integer	\$message:	0x27
version:	1.1.0	\$type:	U32
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

315. Trigger 3 Mode
</mmsp/digitalInput/ioChannel/<index#>/trigger3Mode>

See Figure 8 Digital Input Block Diagram

Get or Set Digital Input [ioChannel] Trigger 3 Mode

0 = Disabled

1 = Low

2 = High

3 = Negative edge (HighToLow)

4 = Positive edge (LowToHigh)

5 = Any edge (HighToLow or LowToHigh)

type:	integer	\$message:	0x28
version:	1.1.0	\$type:	U8
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

316. Trigger 3 Action
</mmsp/digitalInput/ioChannel/<index#>/trigger3Action>

See Figure 8 Digital Input Block Diagram

Get or Set Digital Input [ioChannel] Trigger 3 Action

0 = No action

1 = Turn Emission and EM off

- 2 = Turn EM off
- 3 = Turn Emission on
- 4 = Turn Emission and EM on
- 5 = Turn EM on

type:	integer	\$message:	0x2A
version:	1.1.0	\$type:	U8
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

317. Trigger 3 Action Count

[/mmssp/digitalInput/ioChannel/<index#>/trigger3ActCnt](#)

Get or Set Trigger 3 Action Count

type:	integer	\$message:	0x2B
version:	1.1.0	\$type:	U32
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

318. Trigger 4 Mode

[/mmssp/digitalInput/ioChannel/<index#>/trigger4Mode](#)

See Figure 8 Digital Input Block Diagram

Get or Set Digital Input [ioChannel] Trigger 4 Mode

0 = Disabled

1 = Low

2 = High

3 = Negative edge (HighToLow)

4 = Positive edge (LowToHigh)

5 = Any edge (HighToLow or LowToHigh)

type:	integer	\$message:	0x2C
version:	1.1.0	\$type:	U8
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

319. Trigger 4 Action

[/mmssp/digitalInput/ioChannel/<index#>/trigger4Action](#)

See Figure 8 Digital Input Block Diagram

Get or Set Digital Input [ioChannel] Trigger 4 Action

0 = No action

1 = Turn Emission and EM off

2 = Turn EM off

3 = Turn Emission on

4 = Turn Emission and EM on

5 = Turn EM on

type:	integer	\$message:	0x2E
version:	1.1.0	\$type:	U8
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

320. Trigger 4 Action Count

</mmsp/digitalInput/ioChannel/<index#>/trigger4ActCnt>

Get or Set Trigger 4 Action Count

type:	integer	\$message:	0x2F
version:	1.1.0	\$type:	U32
minimum:	0	\$index:	ioChannel
maximum:	5	\$flags:	NVM PUBLIC
		\$range:	0:5
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

321. Digital Output Parameters

</mmsp/digitalOutput>

type: object \$block: 0x23XX

322. Digital Output Channels

</mmsp/digitalOutput/ioChannelCount>

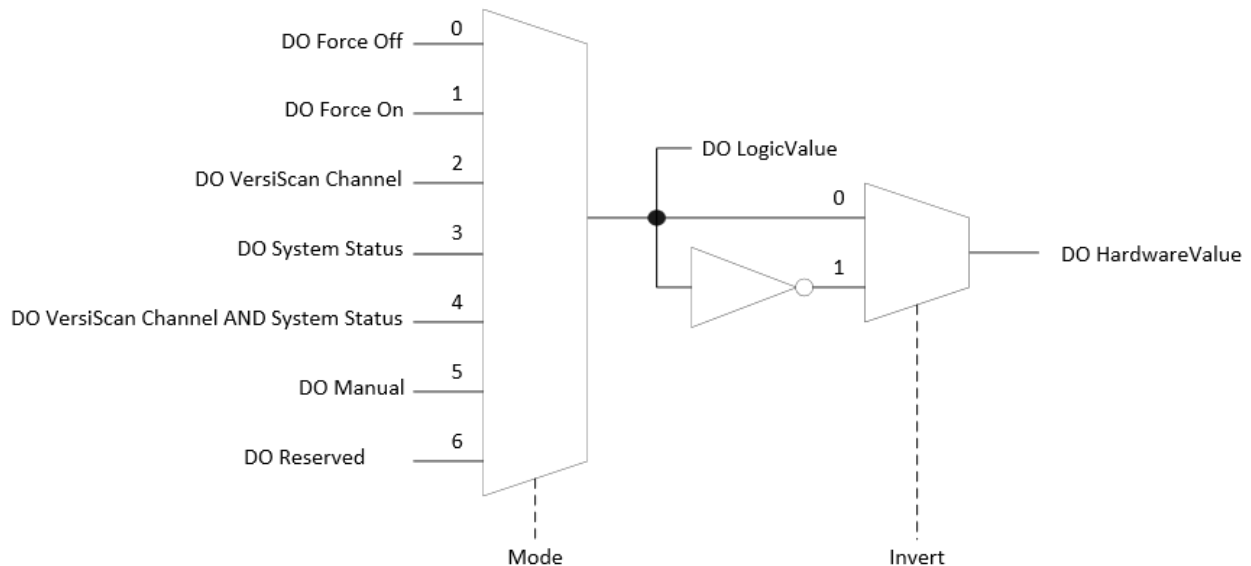


Figure 9 Digital Output Channel Block Diagram

Get the number of possible digital output channels, based on hardware configuration e.g. presence of MultiIO option board.

All PrismaPro MPP electronics come with one relay output.

Access to the relay is through the "AUX I/O" connector pins:

Pin	Description
1	Relay-Common
2	Relay-Normally Open
3	Relay-Normally Closed

This relay is mapped to Digital Output ioChannel 0.

This relay defaults to being energized when the Emission and RF are ON (Mode=3, SS_ALL=0x82000000).

An example of one way to manually Energize/De-energize this relay is shown below.

To energize:

<http://Your PrismaPro MPP IP/mmsp/digitalOutput/ioChannel/0/mode/set?1>

To de-energize:

<http://Your PrismaPro MPP IP/mmsp/digitalOutput/ioChannel/0/mode/set?0>

Additional Digital outputs are available with the multi I/O option board.

```

type:      integer $message:  0x10
version: 0.5.0 $type:        U32
readonly: true $emul:        10
           $flags:          CNST PUBLIC
           $readmode:       normal
           $writemode:      null
           $execmode:       null

```

323. Extended IO Bit Config

[/mmsp/digitalOutput/bitConfig](#)

Get or Set the Extended DIO bits configuration. If a bit is set to zero the corresponding DIO will be an output.

If a bit is set to one the corresponding DIO will be an input. Bits 0 and 1 are not configurable.

Bit(s)	Description
31:18	Reserved (Not Available)
17	Digital Output 17
16	Digital Output 16
15	Digital Output 15
14	Digital Output 14
13	Digital Output 13
12	Digital Output 12
11	Digital Output 11
10	Digital Output 10
9	Digital Output 9
8	Digital Output 8
7	Digital Output 7
6	Digital Output 6
5	Digital Output 5
4	Digital Output 4
3	Digital Output 3
2	Digital Output 2
1	Digital Output 1
0	Digital Output 0

type:	integer	\$message:	0x12
version:	1.4.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

324. Digital Output Default Load

[/mmssp/digitalOutput/defaultLoad](#)

Load and activate the genus specific digital output non volatile parameters.

type:	integer	\$message:	0x18
version:	1.4.0	\$type:	CMD
exec:	true	\$flags:	PUBLIC
writeonly:	true	\$readmode:	null
optional:	true	\$writemode:	null
		\$execmode:	control

325. Digital Output Channels Available

[/mmssp/digitalOutput/channelsAvailable](#)

Get a bit representation of the digital output channels available to the user. The bit definitions are tabulated below. A returned bit of 1 means the channel is available.

Bit(s)	Description
31:19	Reserved (Not Available)
17	Digital Output 17
16	Digital Output 16
15	Digital Output 15
14	Digital Output 14
13	Digital Output 13
12	Digital Output 12
11	Digital Output 11
10	Digital Output 10
9	Digital Output 9
8	Digital Output 8

7	Digital Output 7
6	Digital Output 6
5	Digital Output 5
4	Digital Output 4
3	Digital Output 3
2	Digital Output 2
1	Digital Output 1
0	Digital Output 0

type:	integer	\$message:	0x1A
version:	1.4.2	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

326. Digital Output Manual Values

[/mmssp/digitalOutput/manualValues](#)

Get or Set (service) all Digital Output Manual Values at once, where each bit represents a manual output value.

Bit 0 is value of Manual Value 0

Bit 1 is value of Manual Value 1

...

Bit n is value of Manual Value n

type:	integer	\$message:	0x30
version:	1.0.5	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

327. Digital Output Valve Control Values

</mmsp/digitalOutput/valveControlValues>

Get or Set (service) all Digital Output Valve Control Values at once, where each bit represents a valve control value.

Bit 0 is value of Valve Control Value 0

Bit 1 is value of Valve Control Value 1

...

Bit n is value of Valve Control Value n

type:	integer	\$message:	0x32
version:	1.0.5	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

328. Digital Output Logic Values

</mmsp/digitalOutput/logicValues>

Get or Set (service) all Digital Output Logic Values at once, where each bit represents a logic value.

Bit 0 is value of Logic Value 0

Bit 1 is value of Logic Value 1

...

Bit n is value of Logic Value n

type:	integer	\$message:	0x3E
version:	1.0.5	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

329. Digital Output Hardware Values

</mmsp/digitalOutput/hardwareValues>

Get or Set (service) all Digital Output Hardware Values at once.

Each bit represents a Digital Output value.

The asserted state, represented by a bit value of 1,

indicates the digital output pin is pulled to ground or relay is energized.

The following maps bit value to hardware:

Bit(s)	Description
31:18	Reserved (Not Available)
17	Digital Output 17
16	Digital Output 16
15	Digital Output 15
14	Digital Output 14
13	Digital Output 13
12	Digital Output 12
11	Digital Output 11
10	Digital Output 10
9	Digital Output 9
8	Digital Output 8
7	Digital Output 7
6	Digital Output 6
5	Digital Output 5
4	Digital Output 4
3	Digital Output 3
2	Digital Output 2
1	Digital Output 1
0	Digital Output 0

type:	integer	\$message:	0x40
version:	1.0.5	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

330. Digital Output Native Relay Cycles

</mmsp/digitalOutput/nativeRelayCycles>

Get or Set the number of times the Auxiliary I/O relay opens.

type:	integer	\$message:	0x50
version:	1.2.0	\$type:	U32
		\$flags:	NVM PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

331. ioChannel

</mmsp/digitalOutput/ioChannel>

type:	array	\$range:	0:!IOChannelCount
minIndex:	0		
minItems:	0		
maxItems:	32		

Array of objects with following properties:

type: object

332. Digital Output # Mode

[/mmsp/digitalOutput/ioChannel/<index#>/mode](#)

Get or Set Digital Output Mode (Source Select)

0 = Off

1 = On

2 = VersiScan Channel Threshold Value

3 = System Status Qualified Value

4 = VersiScan Channel AND System Status Qualified Value

5 = Manual Value

6 = Reserved Value

type:	string	\$message:	0x13
version:	0.5.0	\$type:	U8
enum:	Low,High,VersiScanChannel, SystemStatus, VS_SS,Manual,Reserved	\$index:	ioChannel
		\$default:	SystemStatus
		\$flags:	NVM PUBLIC
		\$range:	Low,High,VersiScanChannel,SystemStatus,VS_SS, Manual,Reserved
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

333. Digital Output # Invert

[/mmsp/digitalOutput/ioChannel/<index#>/invert](#)

Get or Set Digital Output[ioChannel] Inversion

0=No Inversion

1=Invert

type:	string	\$message:	0x14
version:	0.5.0	\$type:	U8
enum:	NoInversion,Invert	\$index:	ioChannel
		\$default:	NoInversion
		\$flags:	NVM PUBLIC
		\$range:	NoInversion,Invert
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

334. Digital Output # Logic Value
[/mmssp/digitalOutput/ioChannel/<index#>/logicValue](#)

Get Digital Output [ioChannel] Logic Value

type:	integer	\$message:	0x15
version:	1.0.5	\$type:	U8
		\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

335. Digital Output # Hardware value
[/mmssp/digitalOutput/ioChannel/<index#>/hardwareValue](#)

Get Digital Output [ioChannel] Hardware Value

type:	integer	\$message:	0x16
version:	1.0.5	\$type:	U8
		\$index:	ioChannel
		\$flags:	PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

336. Digital Output # System Status All Not Mask

[/mmsp/digitalOutput/ioChannel/<index#>/ssAllNotMask](#)

Get or Set the "DO_SYSTEM_STATUS_ALL_NOT_MASK" portion of the Digital Output System Status Qualified Value as shown below.

DO_SystemStatusQualifiedValue =

((SystemStatus^DO_SYSTEM_STATUS_ALL_NOT_MASK).AND.DO_SYSTEM_STATUS_ALL_MASK)

== DO_SYSTEM_STATUS_ALL_MASK ||

((SystemStatus^DO_SYSTEM_STATUS_ANY_NOT_MASK).AND.DO_SYSTEM_STATUS_ANY_MASK)

type:	integer	\$message:	0x20
version:	1.0.5	\$type:	U32
		\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

337. Digital Output # System Status All Mask

[/mmsp/digitalOutput/ioChannel/<index#>/ssAllMask](#)

Get or Set the "DO_SYSTEM_STATUS_ALL_MASK" portion of the Digital Output System Status Qualified Value as shown below.

DO_SystemStatusQualifiedValue =

((SystemStatus^DO_SYSTEM_STATUS_ALL_NOT_MASK).AND.DO_SYSTEM_STATUS_ALL_MASK)

== DO_SYSTEM_STATUS_ALL_MASK ||

((SystemStatus^DO_SYSTEM_STATUS_ANY_NOT_MASK).AND.DO_SYSTEM_STATUS_ANY_MASK)

type:	integer	\$message:	0x21
version:	1.0.5	\$type:	U32
		\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

338. Digital Output # System Status Any Not Mask

[/mmsp/digitalOutput/ioChannel/<index#>/ssAnyNotMask](#)

Get or Set the "DO_SYSTEM_STATUS_ANY_NOT_MASK" portion of the Digital Output System Status Qualified Value as shown below.

DO_SystemStatusQualifiedValue =

((SystemStatus^DO_SYSTEM_STATUS_ALL_NOT_MASK).AND.DO_SYSTEM_STATUS_ALL_MASK)


```

== DO_SYSTEM_STATUS_ALL_MASK ||
((SystemStatus^DO_SYSTEM_STATUS_ANY_NOT_MASK).AND.DO_SYSTEM_STATUS_ANY_MASK)

```

type:	integer	\$message:	0x22
version:	1.0.5	\$type:	U32
		\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

339. Digital Output # System Status Any Mask

[/mmsp/digitalOutput/ioChannel/<index#>/ssAnyMask](#)

Get or Set the "DO_SYSTEM_STATUS_ANY_MASK" portion of the Digital Output System Status Qualified Value as shown below.

```

DO_SystemStatusQualifiedValue =
((SystemStatus^DO_SYSTEM_STATUS_ALL_NOT_MASK).AND.DO_SYSTEM_STATUS_ALL_MASK)
== DO_SYSTEM_STATUS_ALL_MASK ||
((SystemStatus^DO_SYSTEM_STATUS_ANY_NOT_MASK).AND.DO_SYSTEM_STATUS_ANY_MASK)

```

type:	integer	\$message:	0x23
version:	1.0.5	\$type:	U32
		\$index:	ioChannel
		\$flags:	NVM PUBLIC
		\$length:	0:!IOChannelCount
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

340. Internal Variables

[/mmisp/internals](#)

type: object \$block: 0x0EXX

341. Boot Count

[/mmisp/internals/bootCount](#)

Number of times this device has been rebooted.

type:	integer	\$message:	0x01
version:	0.17.0	\$type:	U32
		\$flags:	NVM PUBLIC
		\$readmode:	normal
		\$writemode:	normal
		\$execmode:	null

342. Diagnostics

[/mmisp/diagnostics](#)

These targets are used to acquire basic diagnostic information.

type: object \$block: 0x0FXX

343. Anomaly state save trigger conditions

[/mmisp/diagnostics/anomalyStateSaveTriggerConditions](#)

This target defines the requirements for anomalous conditions that cause the system state to be saved into the diagnostics state queue. The bits are logically OR to determined when to trigger.

The bit definitions are tabulated below.

Bit(s)

Description

31

1 = Emission warning

0 = No emission warning

30

1 = Electron multiplier warning

0 = No electron multiplier warning

29

1 = Anode warning

0 = No anode warning

28

1 = RF generator warning

0 = No RF generator warning

27

1 = Internal total pressure warning

0 = No internal total pressure warning

26

1 = Internal power supply warning

0 = No internal power supply warning

25

1 = Electronics box temperature warning

0 = No electronics box temperature warning

24

1 = Electrometer warning

0 = No electrometer warning

23

1 = Database warning (any data block with integrity check)

0 = No Database warning

22:0

Reserved

type:	integer	\$message:	0x40
version:	0.5.0	\$type:	U32
notYetImplemented:	both	\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

344. Gauge

[/mmsp/gauge](#)

type: object \$block: 0x25XX

345. Gauge Control

[/mmsp/gauge/gaugeControl](#)

This target allows the External Active Line gauge to be turned ON or OFF
GET will return the state of the gauge

```
gaugeOverPressure  -1
gaugeUnderPressure -2
gaugeSensorError   -3
gaugeNotPresent     -4
gaugeOFF            0
gaugeInRange        1
```

SET will turn the gauge ON or OFF if there is a gauge Present and return an error status.

Otherwise it will return GaugeNotPresent

type:	number	\$message:	0x10
version:	1.4.2	\$type:	F32
		\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

346. Gauge State

[/mmsp/gauge/gaugeState](#)

This target queries the state of the External Active Line gauge and returns one of the following codes.

```
gaugeOverPressure  -1
gaugeUnderPressure -2
gaugeSensorError   -3
gaugeNotPresent     -4
gaugeOFF            0
gaugeInRange        1
```

type:	integer	\$message:	0x20
version:	1.4.2	\$type:	S16
readonly:	true	\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

347. Gauge Pressure

[/mmsp/gauge/gaugePressure](#)

This target returns the External Active Line gauge Pressure.

The displayed pressure is scaled by the PressureUnits. See the PressureUnits target.

If the gauge is present and in range, it returns the Pressure.

If the gauge is not in range, this function returns the state of the gauge.

```
gaugeOFF          0
gaugeOverPressure -1
gaugeUnderPressure -2
gaugeSensorError  -3
gaugeNotPresent   -4
```

type:	number	\$message:	0x30
version:	1.4.2	\$type:	F32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

348. Gauge Operational Mode

[/mmsp/gauge/gaugeOperMode](#)

This target returns the External Active Line gauge Operation Mode
the function returns

```
0 No known gauge detected
1 Pirani Mode
2 Cold Cathode Mode
```

type:	string	\$message:	0x40
version:	1.4.2	\$type:	STR
notYetImplemented:	set	\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

349. Pressure Units

[/mmsp/gauge/pressureUnits](#)

This target SETs or GETs the displayed pressure units.

The displayed pressure units can be any one of the following:

Torr

mBar

Pascal

If an unknown Units is found, the function returns

Invalid

type:	string	\$message:	0x50
version:	1.4.2	\$type:	STR
		\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

350. Gauge Name

[/mmsp/gauge/gaugeName](#)

This target returns the External Active Line gauge name or NONE if no gauge is present.

PKR360

PKR251

NONE

type:	string	\$message:	0x60
version:	1.4.2	\$type:	STR
notYetImplemented:	set	\$flags:	PUBLIC
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

351. External Gauge Degas Control

[/mmsp/gauge/degas](#)

This command will control the degas function on gauges equipped as such.

Currently, the only gauge supported is the ActiveLine gauge PBR260. Gauge requires 16Watts of power, so the Prisma Pro power supply must be sized accordingly.

type:	integer	\$message:	0x70
version:	1.4.2	\$type:	S8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

XXX. How to following the Links:
[return](#)

API-Reference

PFEIFFER VACUUM

V2018-07-Beta

API Reference

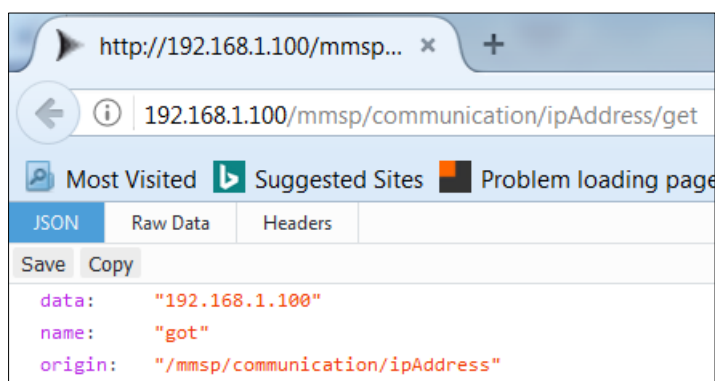
Table of Contents

	Target	Title	/get	/set
1	mmsp	Modular Mass Spec	✓	✓
2	electronicsInfo	Electronics Information	✓	✓
3	serialNumber	Serial Number	✓	✓
4	monitorFWVersion	Monitor Firmware Version	✓	X
5	controlFWVersion	Control Firmware Version	✓	X
6	genus	Genus	✓	✓
7	massRange	Mass Range	✓	X
8	communication	Communications Port Configuration	✓	✓
9	ipAddress	IP Address	✓	✓

10. MMSP MAC Address
[/mmsp/communication/macAddress](#)

MAC Address stored in 1-wire

type:	string	\$message:	0x06
-------	--------	------------	------



(192.168.1.100 = Standard IP-Address after delivery)