

# Documentation

# API Reference PrismaPro

**Pfeiffer Vacuum API Reference List  
for PrismaPro  
(Firmware ≥ 1.9.0)**



Pfeiffer Vacuum  
Berliner Str. 43  
35614 Aßlar, Germany  
Phone: +49 6441 / 802-0  
[QMS@Pfeiffer-Vacuum.de](mailto:QMS@Pfeiffer-Vacuum.de)

**Version from: 2021-04-30**

**Only the API commands specified in this document may be used!**

**You use this API at your own risk!**

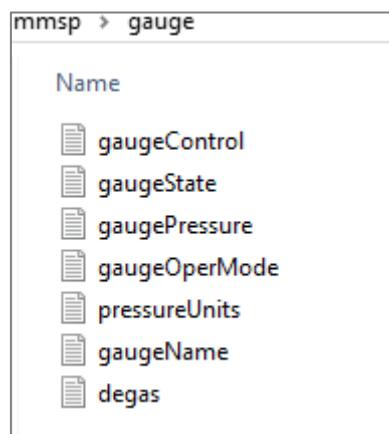
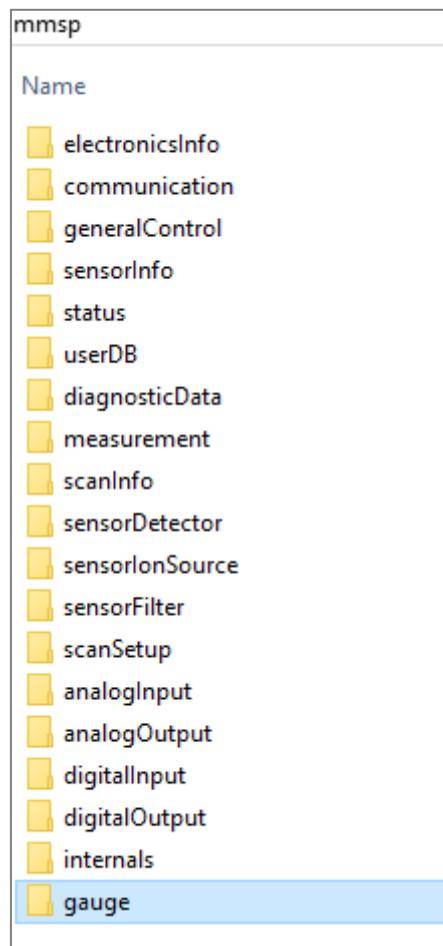


# API Structure

**The API-Commands follow the well-known folder structure from data processing:**

Example: IP-Addr./mmsp/target/..../get

192.168.1.100/mmsp/gauge/gaugePressure/get  
192.168.1.100/mmsp/electronicsInfo/serialNumber/get  
192.168.1.100/mmsp/scanSetup/dwellGlobal/set?128



# API Main Overview

Target	Title	Example
<a href="#">mmsp</a>	<u>Modular Mass Spec</u>	API root directory
<a href="#">electronicsInfo</a>	Electronics Information	Serial nbr, FW-nbr, Mass range
<a href="#">communication</a>	Communications Port Configuration	IP-, MAC- Address, Login, Control info
<a href="#">generalControl</a>	General control	Emission, EM, Degas, User timer
<a href="#">sensorInfo</a>	Sensor Information	Sensor name, Serial number, Ion Source-, Mass Filter-, Detector type
<a href="#">status</a>	Status	System status, Hardware error, Filament
<a href="#">userDB</a>	User Data Base	S5-S8
<a href="#">diagnosticData</a>	Diagnostic Data	Box Temperature, anode-, focus- and filament potential/current,
<a href="#">measurement</a>	Measured Data	Scans, Data
<a href="#">scanInfo</a>	Scan Information	First-/Last -/Current Scan Points per Scan, Scanning
<a href="#">sensorDetector</a>	Sensor Detector parameters	EM-Voltage/Gain, Factory parameters
<a href="#">sensorIonSource</a>	Sensor Ion Source parameters	Anode-, Focus-Min/Max, Electron-, Ion-Energy, Ion source
<a href="#">sensorFilter</a>	Sensor Filter parameters	Mass Min/Max Dwell Min/Max
<a href="#">scanSetup</a>	Versatile Scan parameters	Start-, Stop Channel ppamu, Start- ,Stop mass
<a href="#">analogInput</a>	Analog Input parameters	Channel count Event- Mode, -Action, -Threshold Channel- mode, -bypass, -appValue
<a href="#">analogOutput</a>	Analog Output parameters	Channels available Channel- bypass, -Scale In Mode, -Hardware Value
<a href="#">digitalInput</a>	Digital Input parameters	Channel Count, Hardware Value, Trigger -Mode, -Action, -Count
<a href="#">digitalOutput</a>	Digital Output parameters	Channel available, Hardware Value
<a href="#">internals</a>	Internal Variables	Boot Count
<a href="#">gauge</a>	External Pressure Gauge	Gauge -Pressure, -Name, -Unit

**Total number of API commands for free use: 359**

# Table of Contents

	Target	Title	/get /set
1	<a href="#">mmsp</a>	Modular Mass Spec	✓ ✓
2	<a href="#">electronicsInfo</a>	Electronics Information	✓ ✓
3	<a href="#">serialNumber</a>	Serial Number	✓ ✓
4	<a href="#">monitorFWVersion</a>	Monitor Firmware Version	✓ X
5	<a href="#">controlFWVersion</a>	Control Firmware Version	✓ X
6	<a href="#">genus</a>	Genus	✓ ✓
7	<a href="#">massRange</a>	Mass Range	✓ X
359	<a href="#">rfBoardRightAngle</a>	RF Board Right Angle (FW ≥ 1.9.1)	✓ X
8	<a href="#">communication</a>	Communications Port Configuration	✓ ✓
9	<a href="#">ipAddress</a>	IP Address	✓ ✓
10	<a href="#">macAddress</a>	MMSP MAC Address	✓ ✓
11	<a href="#">gateway</a>	MMSP Gateway	✓ ✓
12	<a href="#">mask</a>	MMSP MAsk	✓ ✓
13	<a href="#">port</a>	MMSP Port	✓ ✓
14	<a href="#">dhcp</a>	Boot from DHCP	✓ ✓
15	<a href="#">sessionID</a>	Session ID	✓ ✓
16	<a href="#">issueLog</a>	Issue Log	✓ ✓
17	<a href="#">errorLog</a>	Error Log	✓ X
18	<a href="#">sessionTimeout</a>	Session Timeout	✓ ✓
19	<a href="#">locationIdentifyCPU</a>	Identify MMSP Location using the CPU Power LED	✓ ✓
20	<a href="#">login</a>	Login	✓ ✓
21	<a href="#">clientName</a>	Client Name	✓ ✓
22	<a href="#">control</a>	Control	✓ ✓
23	<a href="#">controlInfo</a>	Control Info	✓ X
24	<a href="#">sessionID</a>		✓ ✓
25	<a href="#">ipAddress</a>		✓ ✓
26	<a href="#">amInControl</a>		✓ ✓
27	<a href="#">canTakeControl</a>		✓ ✓
28	<a href="#">canForceControl</a>		✓ ✓
29	<a href="#">secondsSinceLastRequest</a>		✓ ✓
30	<a href="#">secondsSinceLastControlRequest</a>		✓ ✓
31	<a href="#">secondsSinceLastDataRequest</a>		✓ ✓
32	<a href="#">clientName</a>		✓ ✓
33	<a href="#">amInControl</a>	Am In Control	✓ X
34	<a href="#">controlLock</a>	ControlLock	✓ ✓
35	<a href="#">generalControl</a>	General control	✓ ✓

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

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

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

	Target	Title	/get /set
153	<a href="#">workingStructure</a>	Working Structure	✓ ✓
154	<a href="#">emissionLevel</a>	Emission Level	✓ ✓
155	<a href="#">optimizationType</a>	Optimization Type	✓ ✓
156	<a href="#">filamentPreferred</a>	Filament Preferred	✓ ✓
157	<a href="#">nomUserCopy</a>	Nominal To User Copy	X ✓
158	<a href="#">nomFactoryCopy</a>	Nominal to Factory Copy	X ✓
159	<a href="#">userTablesValid</a>	User Tables Valid	✓ X
160	<a href="#">factoryTablesValid</a>	Factory Tables Valid	✓ X
161	<a href="#">workingLoadSrc</a>	Working Load Source	✓ X
162	<a href="#">emissionCurrent</a>	Emission current	✓ ✓
163	<a href="#">anode</a>	Anode Voltage	✓ ✓
164	<a href="#">focus</a>	Focus Voltage	✓ ✓
165	<a href="#">electronEnergy</a>	Electron Energy	✓ ✓
166	<a href="#">ionEnergyGlobal</a>	Ion Energy Global	✓ ✓
167	<a href="#">normalizationFactor</a>	Normalization Factor	✓ ✓
168	<a href="#">ppLinConst1</a>	PP Linearization Constant 1	✓ ✓
169	<a href="#">ppLinConst2</a>	PP Linearization Constant 2	✓ ✓
170	<a href="#">ppSensitivityFactor</a>	PP Sensitivity Factor	✓ ✓
171	<a href="#">tPThresholdCurrent</a>	Total Pressure Emission Trip Threshold Current	✓ ✓
172	<a href="#">tPTresholdCount</a>	Total Pressure Emission Trip Threshold Count	✓ ✓
173	<a href="#">tPTripCount</a>	Total Pressure Emission Trip Count	✓ ✓
174	<a href="#">tPEMThresholdCurrent</a>	Total Pressure EM Trip Threshold Current	✓ ✓
175	<a href="#">tPEMThresholdCount</a>	Total Pressure EM Trip Threshold Count	✓ ✓
176	<a href="#">tPEMTripCount</a>	TP EM Trip Count	✓ ✓
177	<a href="#">tPCalPointsMax</a>	Total Pressure Calibration Points Max	✓ X
178	<a href="#">tPCalPointsMin</a>	Total Pressure Calibration Points Min	✓ X
179	<a href="#">tPCalPointsInUse</a>	Total Pressure Calibration Points in Use	✓ ✓
180	<a href="#">tPCalPressSet</a>	Total Pressure Calibration Pressure Set	✓ ✓
181	<a href="#">tPCalPressRemove</a>	Total Pressure Calibration Pressure Remove	✓ ✓
182	<a href="#">tPCalPointRemove</a>	Total Pressure Calibration Point Remove	✓ ✓
183	<a href="#">tPCalTableShift</a>	Total Pressure Calibration Table Shift	✓ ✓

	Target	Title	/get	/set
184	<a href="#">tPunits</a>	TP units	✓	✓
185	<a href="#">degasDuration</a>	Degas Duration	✓	✓
186	<a href="#">emisRestartDelay</a>	Emis Restart Delay	✓	✓
187	<a href="#">emisRestartCount</a>	Emis Restart Count	✓	✓
188	<a href="#">emisRestartType</a>	Emis Restart Type	✓	✓
189	<a href="#">calIndex</a> (array indexed by 0, maxItems: 10)		✓	✓
190	<a href="#">tPCalPointPress</a>	Total Pressure Calibration Point Pressure	✓	✓
191	<a href="#">tPCalPointCurrent</a>	Total Pressure Calibration Point Current	✓	✓
192	<a href="#">tPCalPointSet</a>	Total Pressure Calibration Point Set	✓	✓
193	<a href="#">ionSource</a> (array indexed by 0)		✓	✓
194	<a href="#">defaultLoad</a>	Load Default Parameters	X	✓
195	<a href="#">factoryLoad</a>	Load Factory Parameters	X	✓
196	<a href="#">userLoad</a>	Load User Parameters	X	✓
197	<a href="#">userSave</a>	Save User Parameters	X	✓
198	<a href="#">sensorFilter</a>	Sensor Filter Parameters	✓	✓
199	<a href="#">massMax</a>	Tune Max Mass	✓	X
200	<a href="#">massMin</a>	Tune Min Mass	✓	X
201	<a href="#">dwellMax</a>	Max Allowable Dwell	✓	X
202	<a href="#">dwellMin</a>	Min Allowable Dwell	✓	X
203	<a href="#">designMassMax</a>	Design Max Mass	✓	X
204	<a href="#">designMassMin</a>	Design Min Mass	✓	X
205	<a href="#">rodPolarity</a>	Rod Polarity	✓	✓
206	<a href="#">workingTune</a>	Working Tune	✓	✓
207	<a href="#">maxDwell</a>	Max Allowable Dwell (fixed)	✓	X
208	<a href="#">minDwell</a>	Min Allowable Dwell (fixed)	✓	X
209	<a href="#">peakTopWidenSlope</a>	High Mass Peak Top Widen Slope	✓	✓
210	<a href="#">peakTopWidenIntercept</a>	High Mass Peak Top Widen Intercept	✓	✓
211	<a href="#">peakTopWidenStartMass</a>	Wide Width High Mass - Start Mass	✓	✓
212	<a href="#">scanSetup</a>	Versatile Scan Parameters	✓	✓
213	<a href="#">maxChannels</a>	Versatile Max Scan Channels	✓	X
214	<a href="#">startChannel</a>	Scan Start Channel	✓	✓
215	<a href="#">stopChannel</a>	Scan Stop Channel	✓	✓
216	<a href="#">scanCount</a>	Scan Count	✓	✓
217	<a href="#">scanStart</a>	Scan Start	X	✓
218	<a href="#">scanStop</a>	Scan Stop	✓	✓
219	<a href="#">scanInterval</a>	Scan Interval	✓	✓
220	<a href="#">dwellGlobal</a>	Dwell Global	✓	✓

	Target	Title	/get /set
221	<a href="#">leadInDwell</a>	Lead In Dwell	✓ ✓
222	<a href="#">leakCheckMass</a>	Leak Check Mass	✓ ✓
223	<a href="#">scanTimeTotals</a>	Scan Time Total	✓ X
224	<a href="#">Placeholder</a>	Not currently used	✗ ✗
225	<a href="#">Placeholder</a>	Not currently used	✗ X
226	<a href="#">ppamuRange</a>	ppamu range	✓ X
227	<a href="#">channel</a> (array indexed by 1)		✓ ✓ ✓
228	<a href="#">Placeholder</a>	Not currently used	✗ ✗
229	<a href="#">Placeholder</a>	Not currently used	✗ ✗
230	<a href="#">Placeholder</a>	Not currently used	✗ ✗
231	<a href="#">ppamu</a>	Channel # points per AMU	✓ ✓
232	<a href="#">dwell</a>	Channel # Dwell	✓ ✓
233	<a href="#">emVoltage</a>	Channel # Local EM voltage	✓ ✓
234	<a href="#">Placeholder</a>	Not currently used	✗ ✗
235	<a href="#">Placeholder</a>	Not currently used	✗ ✗
236	<a href="#">extra</a>	Channel # Extra	✓ ✓
237	<a href="#">leadIn</a>	Channel # Lead In	✓ ✓
238	<a href="#">enabled</a>	Channel # Measurement Enabled	✓ ✓
239	<a href="#">scaleFactor</a>	Channel # Scale Factor	✓ ✓
240	<a href="#">equivIonFactor</a>	Channel # Equivalent Ion Current	✓ ✓
241	<a href="#">digOutNum</a>	Channel # Digital Output to Map	✓ ✓
242	<a href="#">digOutThresUpper</a>	Digital Output Upper Threshold	✓ ✓
243	<a href="#">digOutThresLower</a>	Digital Output Lower Threshold	✓ ✓
244	<a href="#">startMass</a>	Channel # Start Mass	✓ ✓
245	<a href="#">stopMass</a>	Channel # Stop Mass	✓ ✓
246	<a href="#">channelMode</a>	Channel Mode	✓ ✓
247	<a href="#">aONum</a>	Channel # Analog Output to Map	✓ ✓
248	<a href="#">aOInputLowLimit</a>	Analog Output Input Low Limit	✓ ✓
249	<a href="#">aOInputHighLimit</a>	Analog Output Input High Limit	✓ ✓
250	<a href="#">aOMode</a>	Analog Output Mode	✓ ✓
251	<a href="#">analogInput</a>	Analog Input Parameters	✓ ✓ ✓
252	<a href="#">ioChannelCount</a>	Analog Input Channels	✓ X
253	<a href="#">defaultLoad</a>	Analog Input Default Load	X ✓
254	<a href="#">eventsDefaultLoad</a>	Analog Input Events Default Load	X ✓
255	<a href="#">channelsAvailable</a>	Analog Input Channels Available	✓ X
256	<a href="#">eventCount</a>	Event Count	✓ X
257	<a href="#">aiEvent</a> (array indexed by 0, maxItems: 32)		✓ ✓ ✓
258	<a href="#">eventMapping</a>	Event to AI mapping	✓ ✓ ✓
259	<a href="#">eventThreshold1</a>	Event Threshold 1	✓ ✓ ✓

	Target	Title	/get /set
260	<a href="#">eventMode1</a>	Event Mode 1	✓ ✓
261	<a href="#">eventAction1</a>	Event Action 1	✓ ✓
262	<a href="#">eventActionCount1</a>	Event Action Count 1	✓ ✓
263	<a href="#">eventThreshold2</a>	Event Threshold 2	✓ ✓
264	<a href="#">eventMode2</a>	Event Mode 2	✓ ✓
265	<a href="#">eventAction2</a>	Event Action 2	✓ ✓
266	<a href="#">eventActionCount2</a>	Event Action Count 2	✓ ✓
267	<a href="#">eventThreshold3</a>	Event Threshold 3	✓ ✓
268	<a href="#">eventMode3</a>	Event Mode 3	✓ ✓
269	<a href="#">eventAction3</a>	Event Action 3	✓ ✓
270	<a href="#">eventActionCount3</a>	Event Action Count 3	✓ ✓
271	<a href="#">eventThreshold4</a>	Event Threshold 4	✓ ✓
272	<a href="#">eventMode4</a>	Event Mode 4	✓ ✓
273	<a href="#">eventAction4</a>	Event Action 4	✓ ✓
274	<a href="#">eventActionCount4</a>	Event Action Count 4	✓ ✓
275	<a href="#">ioChannel</a> (array indexed by 0, maxItems: 32)		✓ ✓
276	<a href="#">rawValue</a>	Analog Input y Raw value	✓ X
277	<a href="#">designOffset</a>	Analog Input y Design Offset	✓ ✓
278	<a href="#">errorOffset</a>	Analog Input y Error Offset	✓ ✓
279	<a href="#">designMultiplier</a>	Analog Input y Design Multiplier	✓ ✓
280	<a href="#">errorMultiplier</a>	Analog Input y Error Multiplier	✓ ✓
281	<a href="#">correctedValue</a>	Analog Input y Corrected value	✓ X
282	<a href="#">mode</a>	Analog Input y Mode	✓ ✓
283	<a href="#">Placeholder</a>	Not currently used	✗ ✗
284	<a href="#">Placeholder</a>	Not currently used	✗ ✗
285	<a href="#">scaledValue</a>	Analog Input y Scaled value	✓ X
286	<a href="#">bypass</a>	Analog Input y Bypass	✓ ✓
287	<a href="#">appValue</a>	Analog Input y Application Value	✓ ✓
288	<a href="#">analogOutput</a>	Analog Output Parameters	✓ ✓
289	<a href="#">ioChannelCount</a>	Analog Output Channels	✓ X
290	<a href="#">defaultLoad</a>	Analog Outputs Default Load	X ✓
291	<a href="#">channelsAvailable</a>	Analog Output Channels Available	✓ X
292	<a href="#">ioChannel</a> (array indexed by 0, maxItems: 32)		✓ ✓
293	<a href="#">raw</a>	Analog Output # Raw	✓ ✓
294	<a href="#">bypass</a>	Analog Output # Bypass	✓ ✓
295	<a href="#">scaleInValue</a>	Analog Output # Scale In Value	✓ ✓
296	<a href="#">scaleMode</a>	Analog Output # Scale Mode	✓ ✓
297	<a href="#">ceiling</a>	Analog Output # Ceiling {Max}	✓ ✓
298	<a href="#">floor</a>	Analog Output # Floor {Min}	✓ ✓

	Target	Title	/get /set
299	<a href="#">scaleOutValue</a>	Analog Out y Scale Out Value	✓ ✓
300	<a href="#">designOffset</a>	Analog Output # Design Offset	✓ ✓
301	<a href="#">errorOffset</a>	Analog Output # Error Offset	✓ ✓
302	<a href="#">designGain</a>	Analog Output # Design Gain	✓ ✓
303	<a href="#">errorGain</a>	Analog Output # Error Gain	✓ ✓
304	<a href="#">hardwareValue</a>	Analog Output # Hardware Value	✓ ✓
305	<a href="#">inputLowLimit</a>	Analog Output # Input Low Limit	✓ ✓
306	<a href="#">inputHighLimit</a>	Analog Output # Input High Limit	✓ ✓
307	<a href="#">digitalInput</a>	Digital Input Parameters	✓ ✓
308	<a href="#">ioChannelCount</a>	Digital Input Channels	✓ X
309	<a href="#">bitConfig</a>	Extended IO Bit Config	✓ ✓
310	<a href="#">rawValueBitwise</a>	Raw Value Bitwise	✓ X
311	<a href="#">defaultLoad</a>	Digital Input Default Load	X ✓
312	<a href="#">channelsAvailable</a>	Digital Input Channels Available	✓ X
313	<a href="#">hardwareValues</a>	Digital Input Hardware Values	✓ X
314	<a href="#">ioChannel</a> (array indexed by 0, maxItems: 32)		✓ ✓
315	<a href="#">rawValue</a>	Digital Input # Raw value	✓ X
316	<a href="#">trigger1Mode</a>	Trigger 1 Mode	✓ ✓
317	<a href="#">trigger1Action</a>	Trigger 1 Action	✓ ✓
318	<a href="#">trigger1ActCnt</a>	Trigger 1 Action Count	✓ ✓
319	<a href="#">trigger2Mode</a>	Trigger 2 Mode	✓ ✓
320	<a href="#">trigger2Action</a>	Trigger 2 Action	✓ ✓
321	<a href="#">trigger2ActCnt</a>	Trigger 2 Action Count	✓ ✓
322	<a href="#">trigger3Mode</a>	Trigger 3 Mode	✓ ✓
323	<a href="#">trigger3Action</a>	Trigger 3 Action	✓ ✓
324	<a href="#">trigger3ActCnt</a>	Trigger 3 Action Count	✓ ✓
325	<a href="#">trigger4Mode</a>	Trigger 4 Mode	✓ ✓
326	<a href="#">trigger4Action</a>	Trigger 4 Action	✓ ✓
327	<a href="#">trigger4ActCnt</a>	Trigger 4 Action Count	✓ ✓
328	<a href="#">digitalOutput</a>	Digital Output Parameters	✓ ✓
329	<a href="#">ioChannelCount</a>	Digital Output Channels	✓ X
330	<a href="#">bitConfig</a>	Extended IO Bit Config	✓ ✓
331	<a href="#">defaultLoad</a>	Digital Output Default Load	X ✓
332	<a href="#">channelsAvailable</a>	Digital Output Channels Available	✓ X
333	<a href="#">manualValues</a>	Digital Output Manual Values	✓ ✓
334	<a href="#">valveControlValues</a>	Digital Output Valve Control Values	✓ ✓
335	<a href="#">logicValues</a>	Digital Output Logic Values	✓ ✓
336	<a href="#">hardwareValues</a>	Digital Output Hardware Values	✓ ✓
337	<a href="#">nativeRelayCycles</a>	Digital Output Native Relay Cycles	✓ ✓

	Target	Title	/get /set
338	<a href="#">ioChannel</a> (array indexed by 0, maxItems: 32)		✓ ✓
339	<a href="#">mode</a>	Digital Output # Mode	✓ ✓
340	<a href="#">invert</a>	Digital Output # Invert	✓ ✓
341	<a href="#">logicValue</a>	Digital Output # Logic Value	✓ ✓
342	<a href="#">hardwareValue</a>	Digital Output # Hardware value	✓ ✓
343	<a href="#">ssAllNotMask</a>	Digital Output # System Status All Not Mask	✓ ✓
344	<a href="#">ssAllMask</a>	Digital Output # System Status All Mask	✓ ✓
345	<a href="#">ssAnyNotMask</a>	Digital Output # System Status Any Not Mask	✓ ✓
346	<a href="#">ssAnyMask</a>	Digital Output # System Status Any Mask	✓ ✓
347	<a href="#">internals</a>	Internal Variables	✓ ✓
348	<a href="#">bootCount</a>	Boot Count	✓ ✓
349	<a href="#">diagnostics</a>	Diagnostics	✓ ✓
350	Placeholder	Not currently used	✗ ✗
351	<a href="#">gauge</a>	Gauge	✓ ✓
352	<a href="#">gaugeControl</a>	Gauge Control	✓ ✓
353	<a href="#">gaugeState</a>	Gauge State	✓ X
354	<a href="#">gaugePressure</a>	Gauge Pressure	✓ X
355	<a href="#">gaugeOperMode</a>	Gauge Operational Mode	✓ ✗
356	<a href="#">pressureUnits</a>	Pressure Units	✓ ✓
357	<a href="#">gaugeName</a>	Gauge Name	✓ ✗
358	<a href="#">degas</a>	External Gauge Degas Control	✓ ✓

### Legend

X Operation not possible.

✓ Operation implemented.

✗ Operation not yet implemented on box.

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

## Appendix:

XXX      [Use Links:](#)

[Standard-IO \(AUX I/O\)](#)  
[Extended-IO](#)

**Example:**  
**How to following the Links**

**Schematic**  
**Schematic**

# 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).

For example, '`<url>/mmsp/sensorIonSource/activate/run`' will execute the api target

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.

## 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:	1.9.0.202010201620

## 2. Electronics Information

### [/mmsp/electronicsInfo](#)

type:	object	\$block:	0x01XX
-------	--------	----------	--------

## 3. Serial Number

### [/mmsp/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:	service
		\$execmode:	null

## 4. Monitor Firmware Version

### [/mmsp/electronicsInfo/monitorFWVersion](#)

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

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'

'DD' is a string indicating the date code

Example: "1.00.00.197001010000" (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.DD"

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'

'DD' is a string indicating the date code

Example: "1.00.00.197001010000" (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. MPxy=1 ... 13, MPP=4

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 properly 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.

#### Communication Port States

type:	object	\$block:	0x02XX
-------	--------	----------	--------

## 359. RF Board Right Angle

NEW NEW NEW NEW: FW1.9.1

### [/mmsp/electronicsInfo/rfBoardRightAngle](#)

Returns TRUE (1) if the RF Board is Right Angle

Go to → [rfBoardRightAngle](#)

## 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

[/mmsp/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

### [/mmsp/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

### [/mmsp/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

### [/mmsp/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

### [/mmsp/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

### [/mmsp/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", "MPP1", ...)

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:	normal
		\$execmode:	null

## 35. General control

### [/mmsp/generalControl](#)

These targets perform general instrument control

type:	object	\$block:	0x04XX
-------	--------	----------	--------

## 36. Set Emission State

### [/mmsp/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

### [/mmsp/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

### [/mmsp/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

### [/mmsp/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

### [/mmsp/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 **and** RF.

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. Wide Width High Mass

[/mmsp/generalControl/peakTopWiden](#)

Enables/Disables tuning RGA with user-defined resolution 0 = User-defined resolution disabled 1 = User-defined resolution enabled

type:	integer	\$message:	0x4C
version:	1.7.0	\$type:	U8
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

## 51. Sensor Information

[/mmsp/sensorInfo](#)

type:	object	\$block:	0x05XX
-------	--------	----------	--------

## 52. Sensor Name

[/mmsp/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

## 53. Sensor Description

[/mmsp/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

## 54. 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

## 55. 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

## 56. 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

## 57. 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

## 58. filaments

### [/mmsp/sensorInfo/filaments](#)

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

**Array** of objects with following properties:

type: object

## 59. 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

## 60. Status

### [/mmsp/status](#)

These targets return the MMSP status

type:	object	\$block:	0x06XX
-------	--------	----------	--------

## 61. System Status

### [/mmsp/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	Reserved
18	0/1 = Peak Top Widen Off/On
17	0/1 = Data Smoothing Off/On
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

## 62. System Status 2

[/mmsp/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

## 63. Hardware errors

### [/mmsp/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

## 64. Filament 1 Open

### [/mmsp/status/fil1Open](#)

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

A Get value of zero indicates filament is good, a value of 1 indicates 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

## 65. Filament 2 Open

### [/mmsp/status/fil2Open](#)

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

A Get value of zero indicates filament is good, a value of 1 indicates 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

## 66. Hardware warnings

### [/mmsp/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

## 67. 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

## 68. 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

## 69. 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

## 70. 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

## 71. CPU Power Stretch

[/mmsp/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

## 72. MotherBoard PowerOnTime

[/mmsp/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

## 73. EM On Time

[/mmsp/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

## 74. EM Pressure Trip

### [/mmsp/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

## 75. filaments

### [/mmsp/status/filaments](#)

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

**Array** of objects with following properties:

type: object

## 76. Emission On Time

### [/mmsp/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

## 77. 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

## 78. 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

## 79. 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

## 80. User DataBase

[/mmsp/userDB](#)

User Database access

type:	object	\$block:	0x07XX
-------	--------	----------	--------

## 81. 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 SDFFileDB.txt

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

## 82. 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 SDFFileDB.txt

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

## 83. S7

### [/mmsp/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 SDFfileDB.txt

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

## 84. S8

### [/mmsp/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 SDFfileDB.txt

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

## 85. Diagnostic Data

### [/mmsp/diagnosticData](#)

These targets return data internal to the MMSP. This data is primarily used for diagnostics

type:	object	\$block:	0x08XX
-------	--------	----------	--------

## 86. Internal Box Temperature

### [/mmsp/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

## 87. Anode potential

### [/mmsp/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

## 88. Emission current

### [/mmsp/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:	µA	\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

## 89. 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

## 90. 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

## 91. 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

## 92. 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

## 93. 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

## 94. 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
-------	--------	----------	--------

## 95. Total Pressure

### [/mmsp/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

## 96. Versatile Scan Data

### [/mmsp/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:

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

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

/mmsp/measurement/scans/0/get returns scan currently in progress.

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

/mmsp/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.

## 97. scannum

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

type:	integer
readonly:	true

## 98. scansize

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

type:	integer
readonly:	true

## 99. values

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

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

Array of number

## 100. 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.

## 101. scannum

### [/mmsp/measurement/scansPow2/<index#>/scannum](#)

type:	integer
readonly:	true

## 102. scansize

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

type:	integer
readonly:	true

## 103. values

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

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

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

## 104. 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

## 105. 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.

## 106. start

[/mmsp/measurement/data/start](#)

type:	integer
readonly:	true

## 107. scansize

[/mmsp/measurement/data/scansize](#)

type:	integer
readonly:	true

## 108. values

### [/mmsp/measurement/data/values](#)

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

Array of number

## 109. Versatile Data - Pow2 Format

### [/mmsp/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

```
value = x * 1e-8 * 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.

### 110. start

[/mmsp/measurement/dataPow2/start](#)

type:	integer
readonly:	true

### 111. scansize

[/mmsp/measurement/dataPow2/scansize](#)

type:	integer
readonly:	true

### 112. values

[/mmsp/measurement/dataPow2/values](#)

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

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

## 113. Versatile Data - Binary Format

[/mmsp/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

## 114. 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.

## 115. start

[/mmsp/measurement/nextScan/start](#)

type:	integer
readonly:	true

## 116. scansize

[/mmsp/measurement/nextScan/scansize](#)

type:	integer
readonly:	true

## 117. values

[/mmsp/measurement/nextScan/values](#)

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

Array of number

## 118. Next Scan Number

### [/mmsp/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

## 119. Binary Next Scan

### [/mmsp/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	0xA0B0C0D
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

## 120. Scan Info

### [/mmsp/scanInfo](#)

Returns the following information about the current scans

type:	object	\$block:	0x0B0X
-------	--------	----------	--------

## 121. First Scan

### [/mmsp/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

## 122. Last Scan

### [/mmsp/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

## 123. Current Scan

### [/mmsp/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

## 124. 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

## 125. 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

## 126. 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 (-1=True, 0=False)
		\$readmode:	locked
		\$writemode:	null
		\$execmode:	null

## 127. Sensor Detector parameters

[/mmsp/sensorDetector](#)

type:	object	\$block:	0x17XX
-------	--------	----------	--------

## 128. 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

## 129. 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

## 130. 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

## 131. 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

## 132. 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

## 133. 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

## 134. Load Factory Parameters

### [/mmsp/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

## 135. Save Factory Parameters

### [/mmsp/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

## 136. Load User Parameters

### [/mmsp/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

## 137. 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

## 138. Sensor Ion Source parameters

### [/mmsp/sensorIonSource](#)

type:	object	\$block:	0x18XX
-------	--------	----------	--------

## 139. Max Emission Current

### [/mmsp/sensorIonSource/emissionCurrentMax](#)

The maximum user settable emission current in microampere ( $\mu\text{A}$ ).

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

## 140. Min Emission Current

### [/mmsp/sensorIonSource/emissionCurrentMin](#)

The minimum user settable emission current in microampere ( $\mu\text{A}$ ).

type:	integer	\$message:	0x21
version:	0.5.0	\$type:	U16
readonly:	true	\$emul:	0
units:	$\mu\text{A}$	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

## 141. Anode Voltage Upper Bound

### [/mmsp/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

## 142. Anode Voltage Lower Bound

### [/mmsp/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

## 143. Max Focus Voltage

### [/mmsp/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

## 144. 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

## 145. 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

## 146. 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

## 147. 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

## 148. 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

## 149. 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

## 150. Activate

### [/mmsp/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

## 151. Selected Filament

### [/mmsp/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

## 152. Selected Filament Override

### [/mmsp/sensorIonSource/filamentSelectedOverride](#)

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

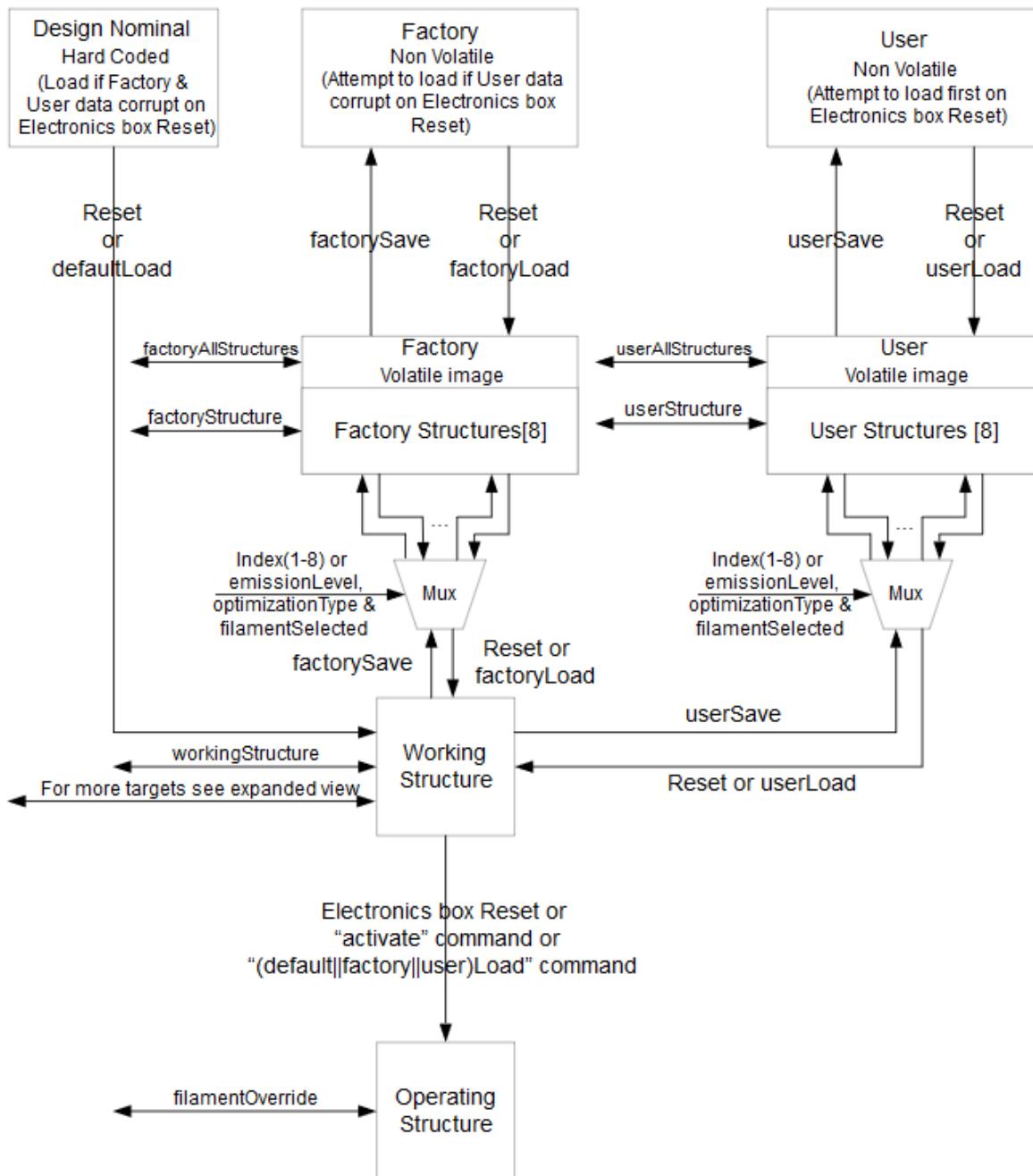
Possible values are 1 and 2.

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

## 153. Working Structure

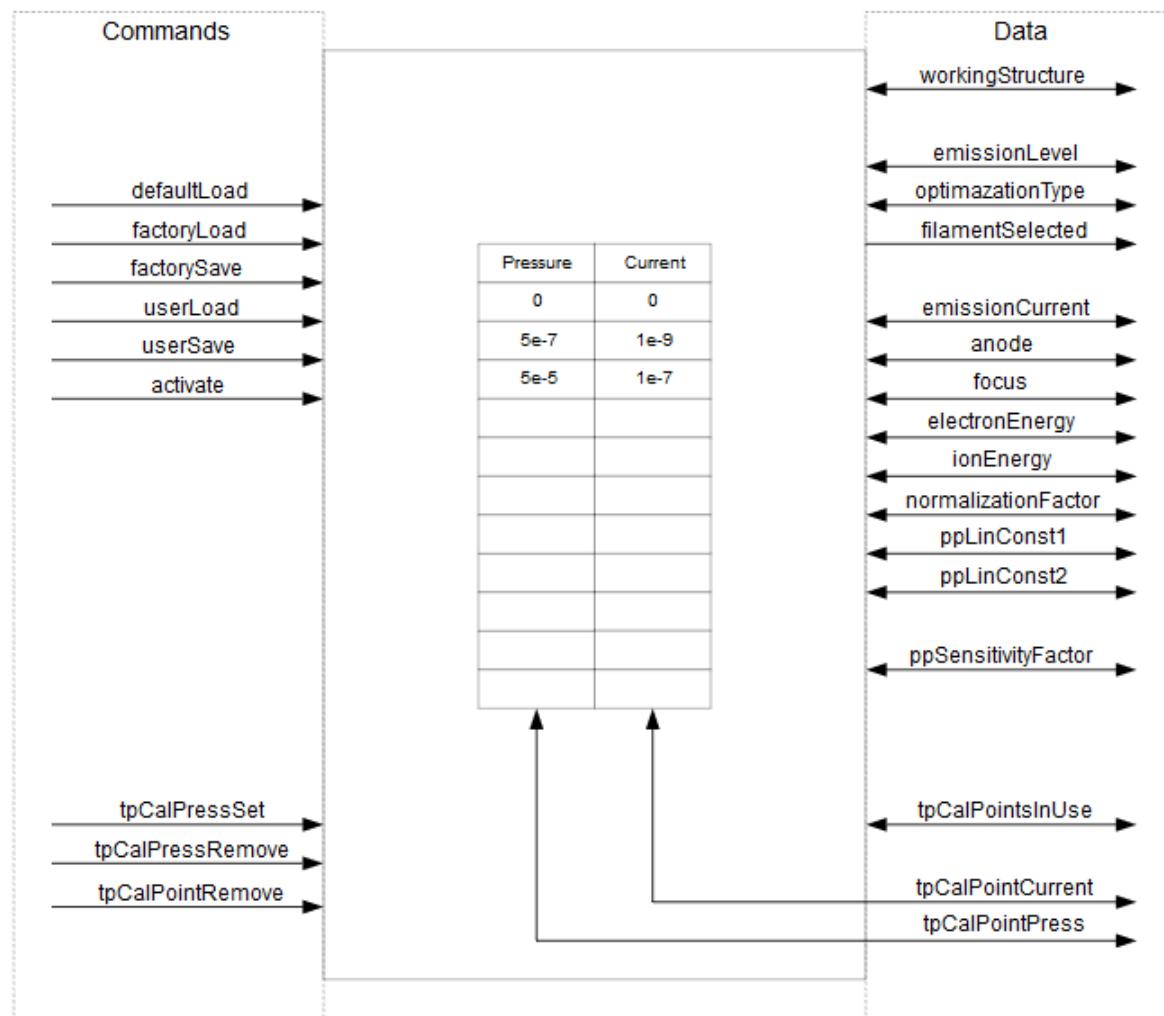
</mmsp/sensorIonSource/workingStructure>

### MMSP sensorIonSource Targets



Sensor Ion Source Targets Full View

## sensorlonSource Targets Working Structure Expanded 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.

## 154. Emission Level

### [/mmsp/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

## 155. Optimization Type

### [/mmsp/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

## 156. Filament Preferred

### [/mmsp/sensorIonSource/filamentPreferred](#)

Get or Set the ion source prefered 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

## 157. Nominal to User Copy

[/mmsp/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

## 158. Nominal to Factory Copy

[/mmsp/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

## 159. User Tables Valid

[/mmsp/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

## 160. Factory Tables Valid

### [/mmsp/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

## 161. 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

## 162. Emission current

### [/mmsp/sensorIonSource/emissionCurrent](#)

Get or Set the working ion source emission current in microampere ( $\mu$ A).

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

## 163. 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

## 164. 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

## 165. 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

## 166. 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

## 167. 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

## 168. 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

## 169. 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

## 170. PP Sensitivity Factor

[/mmsp/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

## 171. Total Pressure Emission Trip Threshold Current

[/mmsp/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

## 172. Total Pressure Emission Trip Threshold Count

[/mmsp/sensorIonSource/tPTresholdCount](#)

Get or Set the ion source 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:	0x71
version:	0.17.0	\$type:	U16
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

## 173. 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

## 174. 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

## 175. 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

## 176. 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

## 177. 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

## 178. 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

## 179. 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

## 180. 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

## 181. 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

## 182. 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

## 183. 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

## 184. 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

## 185. Degas Duration

### [/mmsp/sensorIonSource/degasDuration](#)

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

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

## 186. Emis Restart Delay

### [/mmsp/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

## 187. Emis Restart Count

### [/mmsp/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

## 188. 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

## 189. calIndex

### [/mmsp/sensorIonSource/calIndex](#)

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

**Array** of objects with following properties:

type: object

## 190. 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

## 191. 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

## 192. 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

## 193. ionSource

[/mmsp/sensorIonSource/ionSource](#)

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

**Array** of objects with following properties:

type: object

## 194. Load Default Parameters

[/mmsp/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

## 195. Load Factory Parameters

[/mmsp/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

## 196. 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

## 197. 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

## 198. Sensor Filter Parameters

[/mmsp/sensorFilter](#)

type:	object	\$block:	0x19XX
-------	--------	----------	--------

## 199. 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

## 200. 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

## 201. Max Allowable Dwell

### [/mmsp/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

## 202. Min Allowable Dwell

### [/mmsp/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

## 203. Design Max Mass

### [/mmsp/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

## 204. 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

## 205. 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

## 206. 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.

## 207. Max Allowable Dwell (fixed)

### [/mmsp/sensorFilter/maxDwell](#)

Returns the maximum dwell supported by this device.

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

## 208. Min Allowable Dwell (fixed)

### [/mmsp/sensorFilter/minDwell](#)

Returns the minimum dwell supported by this device.

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

## 209. High Mass Peak Top Widen Slope

### [/mmsp/sensorFilter/peakTopWidenSlope](#)

High Mass Peak Top Widen - Peak Top Widen Slope

type:	integer	\$message:	0xB0
version:	1.7.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

## 210. High Mass Peak Top Widen Intercept

[/mmsp/sensorFilter/peakTopWidenIntercept](#)

High Mass Peak Top Widen - Peak Top Widen Intercept

type:	integer	\$message:	0xB1
version:	1.7.0	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

## 211. Wide Width High Mass - Start Mass

[/mmsp/sensorFilter/peakTopWidenStartMass](#)

High Mass Peak Top Widen - Peak Top Widen Start Mass, value where PTW begins in cAMU

type:	integer	\$message:	0xB2
version:	1.7.0	\$type:	U32
units:	cAMU	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	control
		\$execmode:	null

## 212. Versatile Scan Parameters

### [/mmsp/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
-------	--------	----------	--------

## 213. Versatile Max Scan Channels

### [/mmsp/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

## 214. Scan Start Channel

### [/mmsp/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

## 215. Scan Stop Channel

### [/mmsp/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

## 216. Scan Count

### [/mmsp/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.

Setting the count to 0 or 1: The number of scans will be 1!

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

## 217. Scan Start

### [/mmsp/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

## 218. Scan Stop

### [/mmsp/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

## 219. Scan Interval

### [/mmsp/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

## 220. Dwell Global

### [/mmsp/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

## 221. Lead In Dwell

### [/mmsp/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

## 222. Leak Check Mass

### [/mmsp/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

## 223. Scan Time Total

### [/mmsp/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

## 224. Placeholder

Not currently used

## 225. Placeholder

Not currently used

## 226. ppamu range

### [/mmsp/scanSetup/ppamurange](#)

Returns a list of product specific allowed ppamu values.

type:	string	\$message:	0xC2
version:	1.7.0	\$type:	STR
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

## 227. channel

[/mmsp/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

## 228. Placeholder

**Not currently used**

## 229. Placeholder

**Not currently used**

## 230. Placeholder

**Not currently used**

## 231. 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:	/scanSetup/pamurange
		\$length	340
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

## 232. Channel # Dwell

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

Get or Set a channel's dwell.

Dwell is the amount of time spent taking a reading during 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

**$2^N$  ms, where N is 0:14, and 48 ms**

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

### 233. 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

### 234. Placeholder

**Not currently used**

### 235. Placeholder

**Not currently used**

## 236. 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

## 237. 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

### 238. 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

### 239. 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

## 240. 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

## 241. 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

## 242. 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

## 243. 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

## 244. 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

## 245. 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

## 246. 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	
TotalPressure	Sets Extra and Leadin to 0, Dwell to 2 ms
TotalPressureDetectorCurrent	
CPUBoardTemperature	
ElectrometerPotential	
FixedNumber	

type:	string	\$message:	0x34
version:	0.5.0	\$type:	U16
enum:		\$emul:	1
		\$index:	Channel
		\$flags:	SPECIAL PUBLIC
		\$range:	
		\$readmode:	locked
		\$writemode:	control
		\$execmode:	null

#### 247. Channel # Analog Output to Map

[/mmsp/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

## 248. Analog Output Input Low Limit

[/mmsp/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

## 249. Analog Output Input High Limit

[/mmsp/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

## 250. 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

## 251. Analog Input Parameters

[/mmsp/analogInput](#)

type:	object	\$block:	0x20XX
-------	--------	----------	--------

## 252. Analog Input Channels

[/mmsp/analogInput/ioChannelCount](#)

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

Pin      Name

9          Ain(+)

10        Ain(-)

The JSON target for Analog-In1 is

`http://Your MMSP IP/mmsp/analogInput/ioChannel/1/scaledValue/get`

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

For example, refering to [Figure 6](#), the following JSON target will scale the raw value by 2.0

`http://Your MMSP IP/mmsp/analogInput/ioChannel/1/designMultiplier/set?2.0`

`http://Your MMSP IP/mmsp/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

## 253. Analog Input Default Load

[/mmsp/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

## 254. Analog Input Events Default Load

[/mmsp/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

## 255. Analog Input Channels Available

[/mmsp/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	Pin	
		+	-
31:10	0/1 = Reserved (Not Available)		
9	Analog Input 9 "Extended IO"	49	50
8	Analog Input 8 "Extended IO"	47	48
7	Analog Input 7 "Extended IO"	45	46
6	Analog Input 6 "Extended IO"	43	44
5	Analog Input 5 "Extended IO"	28	29
4	Analog Input 4 "Extended IO"	26	27
3	Analog Input 3 "Extended IO"	24	25
2	Analog Input 2 "Extended IO"	22	23
1	Analog Input 1 "Aux IO" = "Standard IO"	9	10
0	Reserved - Not Available	-	-

## IO-Interface

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

## 256. Event Count

### [/mmsp/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

## 257. aiEvent

### [/mmsp/analogInput/aiEvent](#)

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

**Array** of objects with following properties:

type: object

## 258. Event to AI mapping

### [/mmsp/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

## 259. 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

## 260. 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

## 261. 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

## 262. Event Action Count 1

[/mmsp/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

## 263. Event Threshold 2

</mmsp/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

## 264. 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

## 265. 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

## 266. Event Action Count 2

[/mmsp/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

### 267. Event Threshold 3

[/mmsp/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

### 268. Event Mode 3

[/mmsp/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

## 269. Event Action 3

[/mmsp/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

## 270. Event Action Count 3

[/mmsp/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

## 271. 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

## 272. 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

### 273. 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

### 274. 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

## 275. ioChannel

[/mmsp/analogInput/ioChannel](#)

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

Array of objects with following properties:

type: object

## 276. 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

## 277. 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

## 278. Analog Input y Error Offset

[/mmsp/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

## 279. Analog Input y Design Multiplier

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

Value RawValue multiplied by as a design scaling factor.

In1: Default = 0.002	Standard-IO
In2-9: Default = 1	Extended-IO
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

## 280. 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

## 281. 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

## 282. 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

## 283. Placeholder

**Not currently used**

## 284. Placeholder

**Not currently used**

## 285. 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

## 286. 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

## 287. Analog Input y Application Value

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

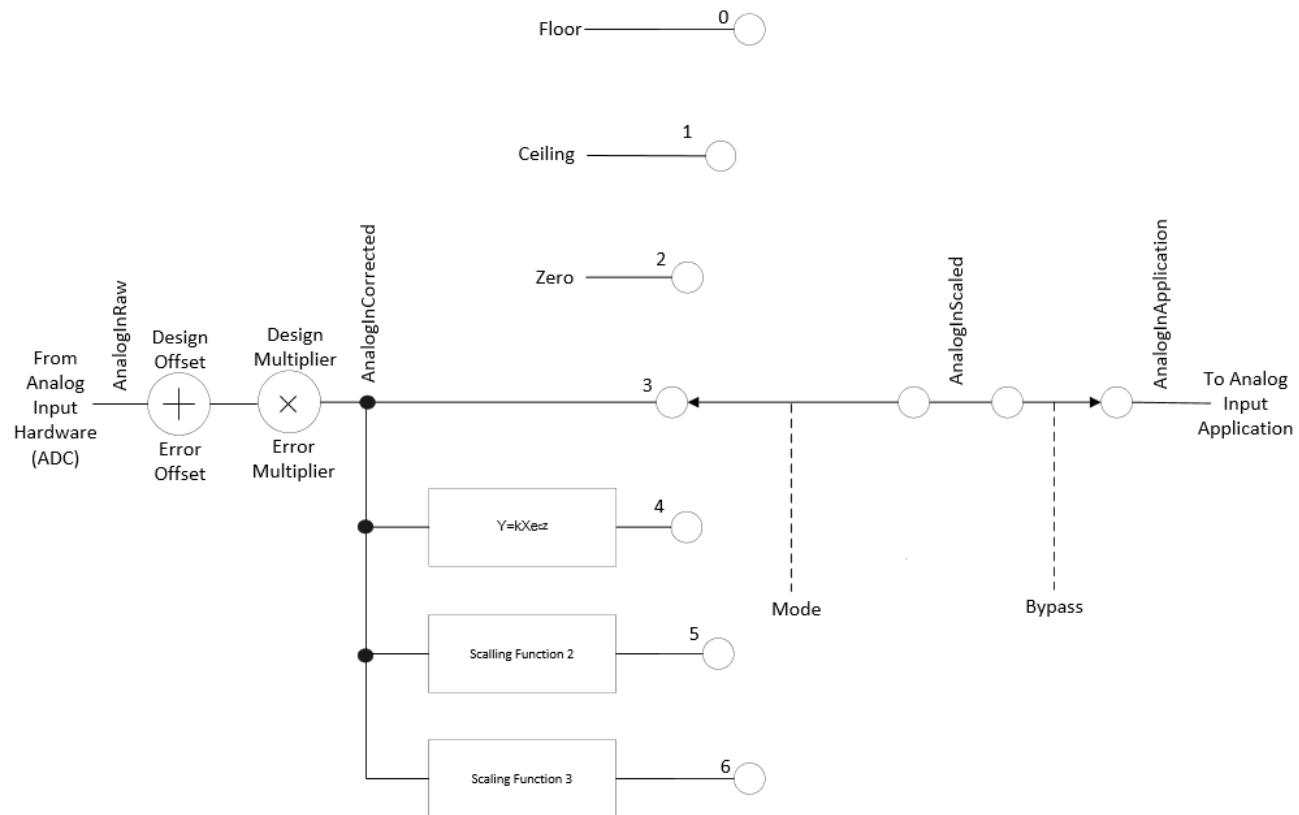


Figure 6 Analog Input Block Diagram

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

## 288. Analog Output Parameters

[/mmsp/analogOutput](#)

type:	object	\$block:	0x21XX
-------	--------	----------	--------

## 289. Analog Output Channels

[/mmsp/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

## 290. Analog Outputs Default Load

[/mmsp/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

## 291. 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	Pin	
		+	-
31:8	Reserved (Not Available)		
7	Analog Output 7 "Extended IO"	38	11
6	Analog Output 6 "Extended IO"	37	13
5	Analog Output 5 "Extended IO"	36	15
4	Analog Output 4 "Extended IO"	35	17
3	Analog Output 3 "Extended IO"	12	55
2	Analog Output 2 "Extended IO"	14	56
1	Analog Output 1 "Extended IO"	16	57
0	Analog Output 0 "Extended IO"	18	58

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

## 292. ioChannel

[/mmsp/analogOutput/ioChannel](#)

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

**Array** of objects with following properties:

type: object

## 293. 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

## 294. 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

## 295. 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

## 296. 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  $C1 + C2 * \log_{10}(\text{Scale In})$

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:

$$\text{Scale Out} = C1 + C2 * \log_{10}(\text{Scale In})$$

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:

$$C2 = \text{Scale Out Max Value} / (\log_{10}(\text{Scale In High Limit}) - \log_{10}(\text{Scale In Low Limit}))$$

$$C1 = -C2 * \log_{10}(\text{Scale In Low Limit})$$

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

## 297. 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

## 298. 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

## 299. 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

### 300. 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

### 301. 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

### 302. 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

### 303. 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

### 304. Analog Output # Hardware Value

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

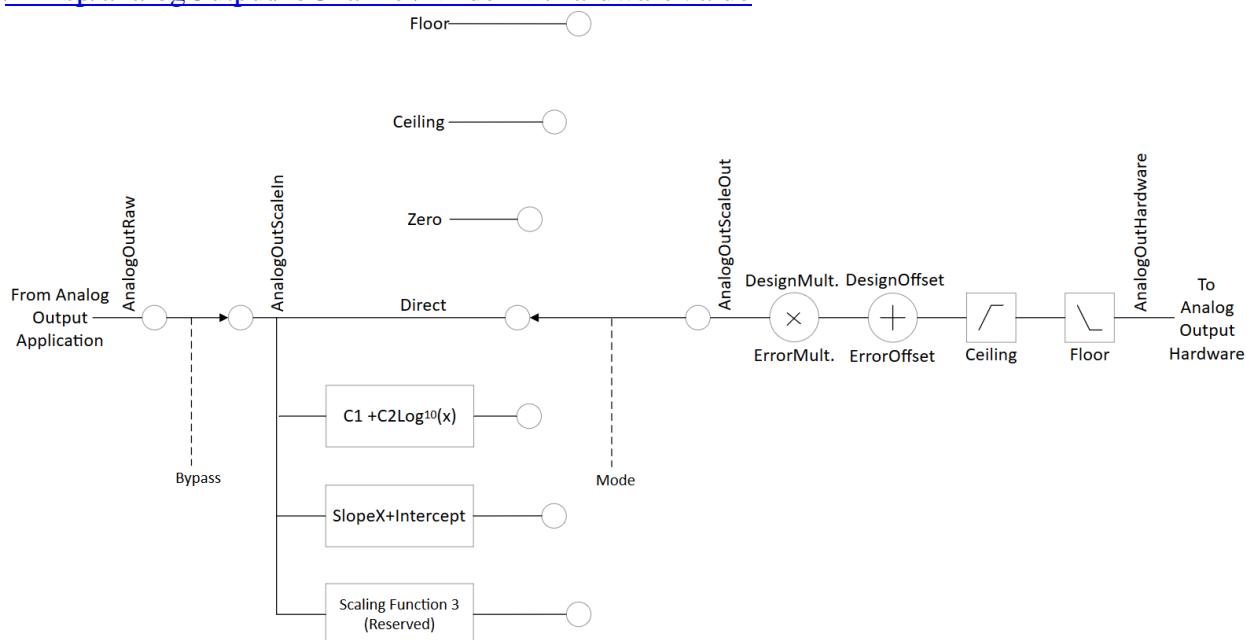


Figure 7 Analog Output Block Diagram

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

### 305. 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

### 306. 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

## 307. Digital Input Parameters

[/mmsp/digitalInput](#)

type:	object	\$block:	0x22XX
-------	--------	----------	--------

## 308. Digital Input Channels

[/mmsp/digitalInput/ioChannelCount](#)

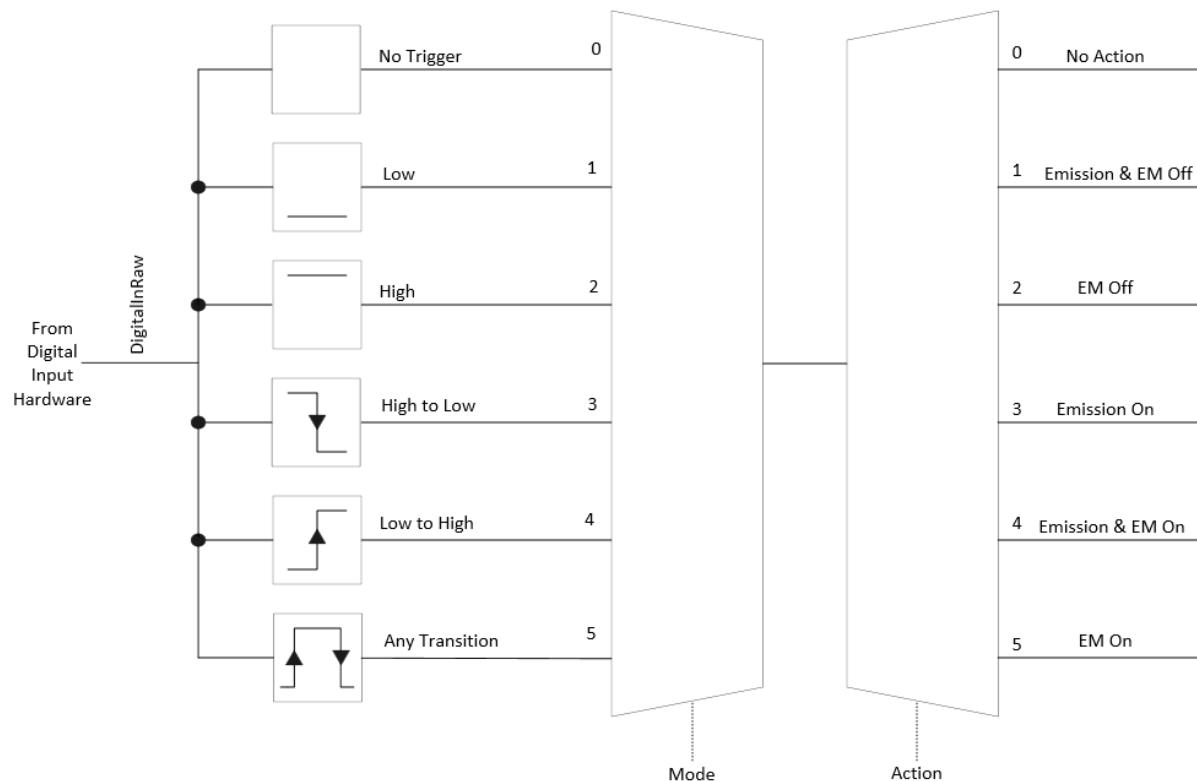


Figure 8 Digital Input Block Diagram

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

### 309. 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.

Input: e.g.: 0011 1110 0000 0000 0011 = dez. 253955

Bit(s)	Description	Pin	
		+	-
31:18	Reserved		
17	Digital Input 17 "Extended IO"	5	6
16	Digital Input 16 "Extended IO"	4	30
15	Digital Input 15 "Extended IO"	3	53
14	Digital Input 14 "Extended IO"	2	54
13	Digital Input 13 "Extended IO"	1	61
12:2	Reserved		
1	Digital Input 1 "Aux IO" = “Standard IO” INPUT ONLY	13	15
0	Digital Input 0 "Aux IO" = “Standard IO” INPUT ONLY	14	

#### [IO-Interface](#)

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

## 310. Raw Value Bitwise

[/mmsp/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	Pin	
		+	-
31:18	Reserved		
17	Digital Input 17 "Extended IO"	5	6
16	Digital Input 16 "Extended IO"	4	30
15	Digital Input 15 "Extended IO"	3	53
14	Digital Input 14 "Extended IO"	2	54
13	Digital Input 13 "Extended IO"	1	61
12:2	Reserved		
1	Digital Input 1 "Aux IO" = “Standard IO” INPUT ONLY	13	15
0	Digital Input 0 "Aux IO" = “Standard IO” INPUT ONLY	14	

[IO-Interface](#)

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

## 311. Digital Input Default Load

[/mmsp/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

## 312. 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.

Input: e.g.: 0011 1110 0000 0000 0011 = dez. 253955

Bit(s)	Description	Pin	
		+	-
31:18	Reserved		
17	Digital Input 17 "Extended IO"	5	6
16	Digital Input 16 "Extended IO"	4	19
15	Digital Input 15 "Extended IO"	3	53
14	Digital Input 14 "Extended IO"	2	54
13	Digital Input 13 "Extended IO"	1	61
12:2	Reserved		
1	Digital Input 1 "Aux IO" = “Standard IO” INPUT ONLY	13	15
0	Digital Input 0 "Aux IO" = “Standard IO” INPUT ONLY	14	

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

### 313. 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	Pin	
		+	-
31:18	Reserved		
17	Digital Input 17 "Extended IO"	5	6
16	Digital Input 16 "Extended IO"	4	30
15	Digital Input 15 "Extended IO"	3	53
14	Digital Input 14 "Extended IO"	2	54
13	Digital Input 13 "Extended IO"	1	61
12:2	Reserved		
1	Digital Input 1 "Aux IO" = “Standard IO” INPUT ONLY	13	15
0	Digital Input 0 "Aux IO" = “Standard IO” INPUT ONLY	14	

type:	integer	\$message:	0x40
version:	1.4.0	\$type:	U32
readonly:	true	\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null

### 314. ioChannel

[/mmsp/digitalInput/ioChannel](#)

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

**Array** of objects with following properties:

type: object

### 315. 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

### 316. 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

### 317. 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

### 318. 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

### 319. 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

### 320. 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

### 321. 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

### 322. 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

### 323. 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

### 324. Trigger 3 Action Count

</mmsp/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

### 325. Trigger 4 Mode

[/mmsp/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

### 326. Trigger 4 Action

[/mmsp/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

**327. 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

## 328. Digital Output Parameters

[/mmsp/digitalOutput](#)

type: object \$block: 0x23XX

## 329. Digital Output Channels

[/mmsp/digitalOutput/ioChannelCount](#)

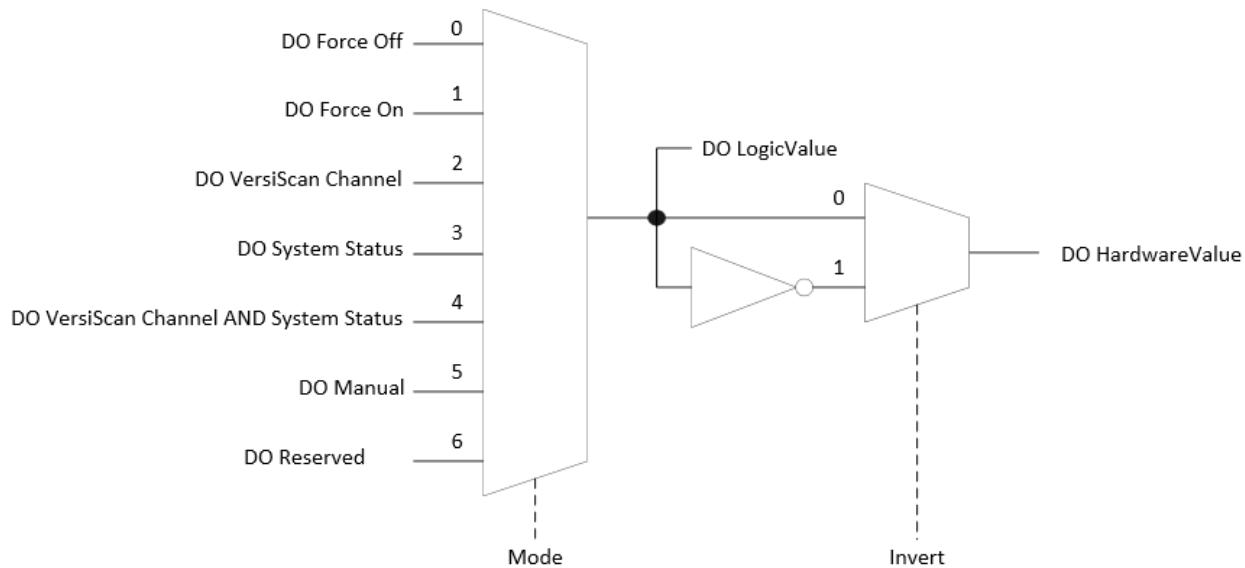


Figure 9 Digital Output Channel Block Diagram

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 IP/mmsp/digitalOutput/ioChannel/0 mode/set?1](http://Your PrismaPro IP/mmsp/digitalOutput/ioChannel/0	mode/set?1)

To de-energize:

[http://Your PrismaPro IP/mmsp/digitalOutput/ioChannel/0 mode/set?0](http://Your PrismaPro 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

### 330. 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	Pin	
		+	-
31:12	Reserved		
11	Digital Output 11 "Extended IO"	34	
10	Digital Output 10 "Extended IO"	33	
9	Digital Output 9 "Extended IO"	32	
8	Digital Output 8 "Extended IO"	31	6
7	Digital Output 7 "Extended IO"	52	19
6	Digital Output 6 "Extended IO"	51	30
5	Digital Output 5 "Extended IO"	10	53
4	Digital Output 4 "Extended IO"	9	54
3	Digital Output 3 "Extended IO"	8	61
2	Digital Output 2 "Extended IO"	7	
1	Digital Output 1 Reserved (Not Available)	-	
0	Digital Output 0 "Aux IO" = <b>Standard-IO</b> Relay pin 1(COM) pin 2(NO) pin 3(NC) OUTPUT ONLY	1 2 3	15

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

## 331. Digital Output Default Load

### [/mmsp/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

## 332. Digital Output Channels Available

### [/mmsp/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	Pin	
		+	-
31:7	Reserved		
11	Digital Output 11 "Extended IO"	34	
10	Digital Output 10 "Extended IO"	33	
9	Digital Output 9 "Extended IO"	32	
8	Digital Output 8 "Extended IO"	31	6
7	Digital Output 7 "Extended IO"	52	19
6	Digital Output 6 "Extended IO"	51	30
5	Digital Output 5 "Extended IO"	10	53
4	Digital Output 4 "Extended IO"	9	54
3	Digital Output 3 "Extended IO"	8	61
2	Digital Output 2 "Extended IO"	7	
1	Digital Output 1 Reserved (Not Available)	-	
0	Digital Output 0 "Aux IO" = <b>Standard-IO</b> Relay pin 1(COM) pin 2(NO) pin 3(NC) OUTPUT ONLY	1 2 3	15

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

### 333. Digital Output Manual Values

[/mmsp/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

### 334. 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

## 335. 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

## 336. 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	Pin	
		+	-
31:7	Reserved		
11	Digital Output 11 "Extended IO"	34	
10	Digital Output 10 "Extended IO"	33	
9	Digital Output 9 "Extended IO"	32	
8	Digital Output 8 "Extended IO"	31	6
7	Digital Output 7 "Extended IO"	52	19
6	Digital Output 6 "Extended IO"	51	30
5	Digital Output 5 "Extended IO"	10	53
4	Digital Output 4 "Extended IO"	9	54
3	Digital Output 3 "Extended IO"	8	61
2	Digital Output 2 "Extended IO"	7	
1	Digital Output 1 Reserved (Not Available)	-	
0	Digital Output 0 "Aux IO" = <b>Standard-IO</b> Relay pin 1(COM) pin 2(NO) pin 3(NC) OUTPUT ONLY	1 2 3	15

type:	integer	\$message:	0x40
version:	1.0.5	\$type:	U32
		\$flags:	PUBLIC
		\$readmode:	normal
		\$writemode:	service
		\$execmode:	null

### 337. 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

### 338. ioChannel

[/mmsp/digitalOutput/ioChannel](#)

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

**Array** of objects with following properties:

type: object

### 339. 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

### 340. 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

### 341. Digital Output # Logic Value

[/mmsp/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

### 342. Digital Output # Hardware value

[/mmsp/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

### 343. 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

### 344. Digital Output # System Status All Mask

[/mmsp/digitalOutput/ioChannel/<index#>/ssAllMask](#)

Get or Set the "DO\_SYSTEM\_STATUS\_ALL\_MASK" portion of the Digital Ouput 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

### 345. 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 Ouput 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

### 346. Digital Output # System Status Any Mask

[/mmsp/digitalOutput/ioChannel/<index#>/ssAnyMask](#)

Get or Set the "DO\_SYSTEM\_STATUS\_ANY\_MASK" portion of the Digital Ouput 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

## 347. Internal Variables

[/mmsp/internals](#)

type:	object	\$block:	0x0EXX
-------	--------	----------	--------

## 348. Boot Count

[/mmsp/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:	admin
		\$execmode:	null

## 349. Diagnostics

[/mmsp/diagnostics](#)

These targets are used to acquire basic diagnostic information.

type:	object	\$block:	0x0FXX
-------	--------	----------	--------

## 350. Placeholder

**Not currently used**

## 351. Gauge

[/mmsp/gauge](#)

type:	object	\$block:	0x25XX
-------	--------	----------	--------

## 352. 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

## 353. 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

## 354. 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

## 355. 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

## 356. 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

## 357. 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

## 358. External Gauge Degas Control

### [/mmsp/gauge/degas](#)

This command will control the degas function on gauges equiped 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

## 359. RF Board Right Angle

NEW NEW NEW NEW

</mmsp/electronicsInfo/rfBoardRightAngle>

Returns TRUE (1) if the RF Board is Right Angle

type:	boolean	\$message:	0x45
version:	1.9.1	\$type:	BOOL
readonly:	true	\$flags:	CNST PUBLIC
		\$readmode:	normal
		\$writemode:	null
		\$execmode:	null



[top](#)

## Appendix:

### XXX. How to following the Links:

		Target	Title	/get /set
1	<a href="#">mmsp</a>		Modular Mass Spec	✓ ✓
2	<a href="#">electronicsInfo</a>		Electronics Information	✓ ✓
3	<a href="#">serialNumber</a>		Serial Number	✓ ✓
4	<a href="#">monitorFWVersion</a>		Monitor Firmware Version	✓ X
5	<a href="#">controlFWVersion</a>		Control Firmware Version	✓ X
6	<a href="#">genus</a>		Genus	✓ ✓
7	<a href="#">massRange</a>		Mass Range	✓ X
8	<a href="#">communication</a>		Communications Port Configuration	✓ ✓
9	<a href="#">ipAddress</a>		IP Address	✓ ✓

### 9. IP Address

</mmsp/communication ipAddress>

IP Addr <http://192.168.1.100/mmsp/communication/ipAddress/get>

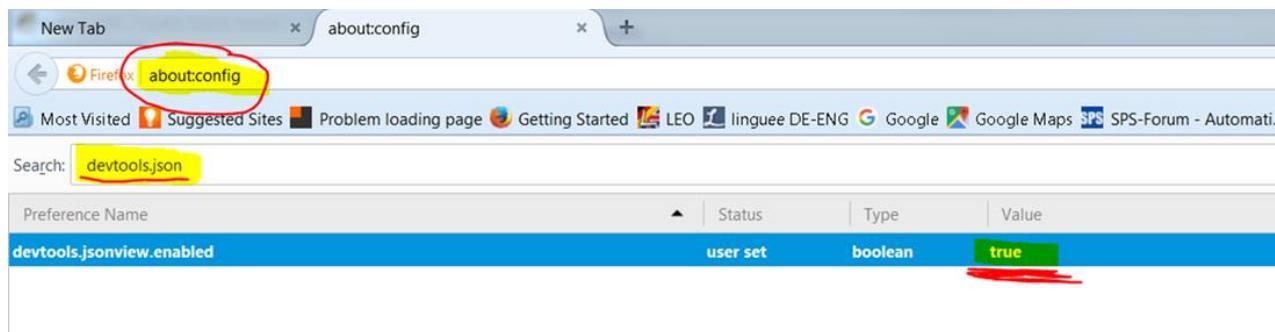
type:	string	\$message:	0x05
version:	0.5.0	\$type:	STR
minLength:	0	\$emul:	192.168.1.1

```

http://192.168.1.100/mmsp...
192.168.1.100/mmsp/communication/ipAddress/get
data: "192.168.1.100"
name: "got"
origin: "/mmsp/communication/ipAddress"
  
```

(192.168.1.100 = Standard IP-Address after delivery)

## Use JSONVIEW with Firefox:



### \$modes:

(\$readmode, \$writemode, &execmode)

\$readmode:	normal
\$writemode:	control
\$execmode:	null

*null, locked, normal, control, protected, admin, service*

**null:**

**No function**

**normal:**

**Without restrictions**

**control:**

**Client must request the control token**

Non-Controlling sessions are still able to read data

**locked:**

**Is an elevated form of control mode.**

If the PrismaPro is locked, controlling session would no longer be able to read and will have very limited access to the information (only static data such serial number)

**protected:**

**Username and Password are required!**

**Login as EULA\* -User**

**admin, service:**

**Username and Password are required!**

**When using an API in “protected” mode, the EULA\* must be signed!**

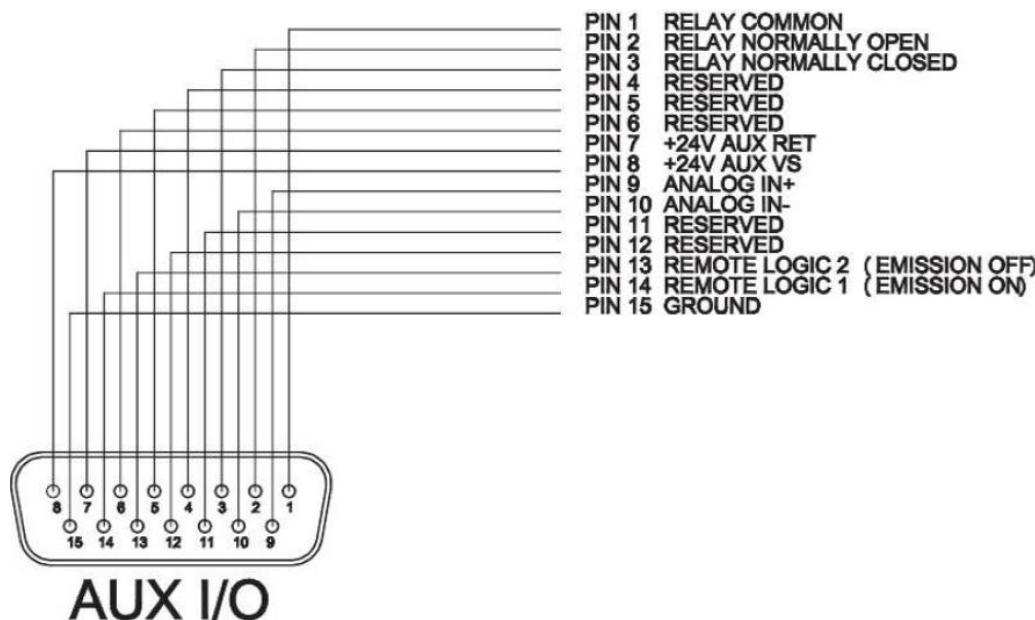
*(“admin” and “service” are reserved for the manufacturer)*

*Ask for the EULA\* document:*

**QMS@Pfeiffer-Vacuum.de**

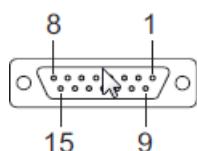
\*End User License Agreement between the Customer and Pfeiffer Vacuum

## Standard-IO (AUX I/O): [\(PrismaPro IO-Interface\)](#)



The <AUX I/O> connection is comprised of a 15-pin D-Sub socket.

The PrismaPro electronic unit QME supports I/O functions via the <AUX I/O> connection on the front panel.



**Fig. 10: AUX I/O (15-pin D-Sub socket)**

1	Relay (COMMON)	10	Analog output 1 (-)
2	Relay (NO, normally opened)	13	Digital input 1 (pre-setting: Emission OFF)
3	Relay (NC, normally closed)	14	Digital input 0 (pre-setting: Emission ON)
7	0 V (GND for 24 V output)	15	GND
8	+ 24 V (output, max. 1 A)	4, 5, 6, 11, 12	unassigned
9	Analog output 1 (+)		

### **Relay**

The status relay output is active (closed) when the emission is switched on.

- Emission ON: Pin 1 + Pin 2 connected = Relay closed.
- Emission OFF: Pin 1 + Pin 2 not connected = Relay open.
- Contact load: 24 V (DC) at 0.5 A

## Extended-IO (EXT I/O):

Extended control connection EXT I/O (optional)

Plug-in connector	HD D-Sub, 62-pin, female	
Analog outputs	8x	0 V - +10 V, output resistance 100 Ω, I <sub>max</sub> = 10 mA, resolution 16 bit
Analog inputs	8x	-10 V - +10 V, input resistance 50 MΩ, resolution 16 bit
Digital outputs	10x	Nominal +24 V, max. 28 V, 200 mA
Digital inputs	5x	Low < +2 V; High > +3 V, nominal 24 V
Output voltage	+24 V, max. 1 A (including current to +24 V output at standard control connection)	

The Extended-IO connection is comprised of a 62-pin HD D-Sub socket.

The PrismaPro electronic unit QME supports extended I/O functions via the Standard-IO connection on the front panel if the optional IO250 is installed



**Fig. 11: EXT I/O (62-pin HD D-Sub socket)**

Pin		Pin	
1	Digital Input 13	32	Digital Output 9
2	Digital Input 14	33	Digital Output 10
3	Digital Input 15	34	Digital Output 11
4	Digital Input 16	35	Analog Output 4
5	Digital Input 17	36	Analog Output 5
7	Digital Output 2	37	Analog Output 6
8	Digital Output 3	38	Analog Output 7
9	Digital Output 4	21,39,40,60	+24 V <sup>2)</sup>
10	Digital Output 5	20,41,42,62	0 V <sup>3)</sup>
12	Analog Output 3	43	Analog Input 6 (+)
14	Analog Output 2	44	Analog Input 6 (-)
16	Analog Output 1	45	Analog Input 7 (+)
18	Analog Output 0	46	Analog Input 7 (-)
22	Analog Input 2 (+)	47	Analog Input 8 (+)
23	Analog Input 2 (-)	48	Analog Input 8 (-)
24	Analog Input 3 (+)	49	Analog Input 9 (+)
25	Analog Input 3 (-)	50	Analog Input 9 (-)
26	Analog Input 4 (+)	51	Digital Output 6
27	Analog Input 4 (-)	52	Digital Output 7
28	Analog Input 5 (+)	6,19,30,53,54,61	GND_Signal
29	Analog Input 5 (-)	11,13,15,17,55,56,57,58,59	Analog Mass (Analog return)
31	Digital Output 8		

<sup>2)</sup> The 24 V supply voltage is available at all Pins +24 V / 0 V for switching with digital inputs / outputs.  
Together, a maximum of 1 A current may flow through.

<sup>3)</sup> The 24 V supply voltage is available at all Pins +24 V / 0 V for switching with digital inputs / outputs.  
Together, a maximum of 1 A current may flow through.

**Fig. 12 Schematic Extended Connector:**

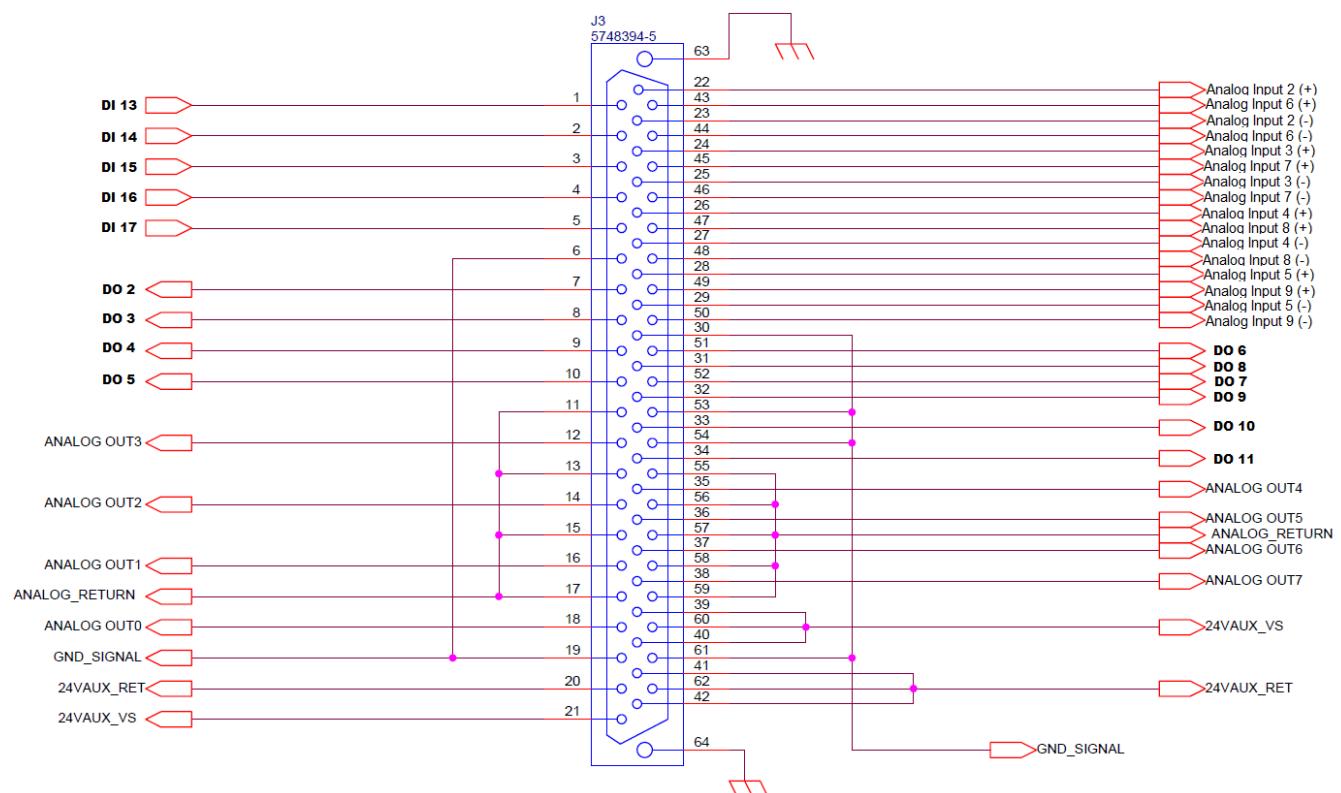


Table 1 - Analog Outputs - Pin Assignments

Designation	DB62 connector Pin
Analog Output 0	18
Analog Output 1	16
Analog Output 2	14
Analog Output 3	12
Analog Output 4	35
Analog Output 5	36
Analog Output 6	37
Analog Output 7	38
Analog Return	11,13,15,17,55,56,57,58,59

Table 2 - Analog Inputs - Pin Assignments

Designation	DB62 connector Pin	Designation	DB62 connector Pin
Analog Input 2(+)	22	Analog Input 2(-)	23
Analog Input 3(+)	24	Analog Input 3(-)	25
Analog Input 4(+)	26	Analog Input 4(-)	27
Analog Input 5(+)	28	Analog Input 5(-)	29
Analog Input 6(+)	43	Analog Input 6(-)	44
Analog Input 7(+)	45	Analog Input 7(-)	46
Analog Input 8(+)	47	Analog Input 8(-)	48
Analog Input 9(+)	49	Analog Input 9(-)	50

Table 3 - Digital Outputs - Pin Assignments

Designation	DB62 connector Pin
Digital Output 2	7
Digital Output 3	8
Digital Output 4	9
Digital Output 5	10
Digital Output 6	51
Digital Output 7	52
Digital Output 8	31
Digital Output 9	32
Digital Output 10	33
Digital Output 11	34
GND_SIGNAL	6,19,30,53,54,61

Table 4 - Digital Inputs - Pin Assignments

Designation	DB62 Connector Pin
Digital Input 13	1
Digital Input 14	2
Digital Input 15	3
Digital Input 16	4
Digital Input 17	5
GND_SIGNAL	6,19,30,53,54,61

