

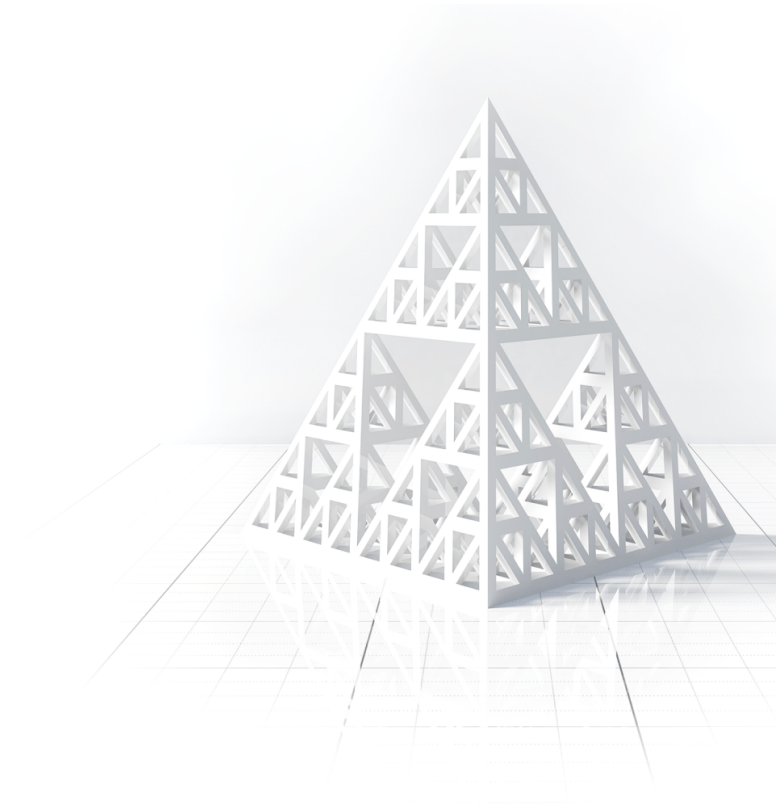
**PYRAMID**  
www.pyramid.tech

IG2

# **IG2 - User Manual**

Document ID: 2614658216

Version: v4



## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Revisions	4
1.2	References	4
<b>2</b>	<b>Document Approvals</b>	<b>5</b>
2.1	Signatures	5
<b>3</b>	<b>Overview</b>	<b>6</b>
3.1	What is EPICS?	6
3.2	How does IG2 use EPICS?	6
3.3	What EPICS software can be used with IG2?	6
3.4	Supported Device Types and Wires	7
3.5	IG2 Usage and Startup	7
<b>4</b>	<b>IG2 Configuration and Channels</b>	<b>8</b>
4.1	Channel Scaling	9
4.2	Channel Limits	9
4.3	Channel Buffering	9
4.4	Monitor Only Change	9
4.5	Realtime Processor Configuration File	10
<b>5</b>	<b>Pyramid Devices</b>	<b>11</b>
5.1	Supported Device Types and Wires	11
5.2	Overview	11
5.3	A560	11
5.4	B10	12
5.5	C400	13
5.6	F100	14
5.7	F3200E	15
5.8	F460	16
5.9	H10	20
5.10	H20	20
5.11	I128	22
5.12	I200	27
5.13	I3200	29
5.14	I400	29

5.15 IC101 .....	30
5.16 M10 .....	32
5.17 M40.....	32
5.18 N2400.....	33
<b>6 Virtual Devices .....</b>	<b>34</b>
6.1 ASCIIClient .....	34
6.2 BPM (Detector).....	34
6.3 GAUSSIANFIT (Detector).....	35
6.4 GCH30.....	36
6.5 ILB ETH 24 DI16 DIO16-2TX.....	36
6.6 IONCHAMBER (Detector) .....	37
6.7 KEITHLEY .....	38
6.8 KOLLMORGEN (MODBUS).....	38
6.9 KOLLMORGEN (TELNET).....	39
6.10 MEMBLOCK.....	40
6.11 PIDCONTROL.....	40
<b>7 Appendix 1 - Advanced Channel Buffering .....</b>	<b>42</b>
7.1 Design.....	42
7.2 Global Options .....	42
7.3 Reading Timeslice Database from RTP file.....	42

# 1 Introduction

**Document ID:** 2614658216

## 1.1 Revisions

Version	Description	Saved by	Saved on
v4	Removed outdated references	Duyen Nguyen	Aug 11, 2025 10:25 AM
v3.2	Modified some I128 wires.	William Nett	Apr 12, 2024 8:04 PM
v3.1	Imported from PTC2-2120709629-5003 from Sep 8, 2021	William Nett	Apr 12, 2024 5:50 PM

## 1.2 References

Document	Document Id	Author	Version

## 2 Document Approvals

This document has been reviewed and approved as follows.



**Document Control** Not Reviewed

Current document version: v.1

No reviewers assigned.

### 2.1 Signatures

for most recent document version

Monday, Aug 11, 2025, 10:30 AM UTC

**Phan Thi Thao Nguyen signed ; meaning: Review**

Monday, Aug 11, 2025, 10:30 AM UTC

**Duyen Nguyen signed ; meaning: Review**

## 3 Overview

### 3.1 What is EPICS?

From the EPICS website (<http://www.aps.anl.gov/epics/index.php>):

“EPICS is a set of software tools and applications which provide a software infrastructure for use in building distributed control systems to operate devices such as Particle Accelerators, Large Experiments and major Telescopes. Such distributed control systems typically comprise tens or even hundreds of computers, networked together to allow communication between them and to provide control and feedback of the various parts of the device from a central control room, or even remotely over the internet.

EPICS uses Client/Server and Publish/Subscribe techniques to communicate between the various computers. Most servers (called Input/Output Controllers or IOCs) perform real-world I/O and local control tasks, and publish this information to clients using the Channel Access (CA) network protocol. CA is specially designed for the kind of high bandwidth, soft real-time networking applications that EPICS is used for, and is one reason why it can be used to build a control system comprising hundreds of computers.”

### 3.2 How does IG2 use EPICS?

IG2 includes an optional Channel Access Server (CAS), or Portable Server, that implements EPICS Channel Access Protocol. This allows any EPICS client software to readback and control IG2 I/O points using EPICS Process Variables (PVs).

IG2 has a database of named I/O points, called channels, listed in its configuration file (default: system.xml). Each of these channels has an associated type and direction. These channels map one-to-one to EPICS PVs made available by IG2. This list of channels represents the full list of EPICS PVs available.

This hardware configuration file contains a section for virtual devices that allows the EPICS CAS to be enabled in IG2. To enable the EPICS CAS in IG2, add the following <epicscas> node to the <interpreter> section of the file.

```
### system.xml

<interpreter>

  <devices>

    <epicscas type="epicscas" name="epics_server" />

  </devices>

</interpreter>

###
```

### 3.3 What EPICS software can be used with IG2?

Almost any EPICS software tool can be used with IG2. There is a vast selection of canned tools and programming APIs available. Many of them can be found on the EPICS website: <http://www.aps.anl.gov/epics/extensions/index.php>. There is support for C/C++, Java, Matlab, Perl, Python, C#, PHP, and other languages. There are canned tools for backup/restore, archiving, plotting, and monitoring.

*Any EPICS client that need to communicate with IG2 must be in the same network broadcast domain as IG2.*

### **3.4 Supported Device Types and Wires**

The system.xml hardware configuration is separated into 2 main sections, or nodes. The first is <loopcontrollers>. This section contains a complete listing of standard Pyramid hardware devices in the system. The other section is called <interpreter>. This section contains a listing of custom hardware and virtual devices. Sections 6 and 7 below list supported device types and their associated i/o (wires).

### **3.5 IG2 Usage and Startup**

IG2 is a console application that takes one command line argument for the path of the xml configuration file. Example:

**ig2-2.0.2 c:\config\mysystem.xml**

The configuration file path argument is optional. IG2 will use "system.xml" by default, located in the local application directory.

## 4 IG2 Configuration and Channels

The system.xml file contains a listing of the hardware configuration that IG2 will manage. This configuration is user-specific. IG2 offers a set of device types that can be controlled. A set of I/O points, called wires, is available within an instance of each of these device types. Users can define an arbitrary number of channels within the device instance which map to a specified wire to provide control and readback.

```
### system.xml

<board type="M10" name="XQ7_ctrl" address="7">

  <channels>

    <channel name="c_XQ7_current_ctrl" wire="analog_out_1" limitLow="-10"
limitHigh="10" />

<channel name="r_XQ7_current_ctrl" wire="analog_in_1" scaleB="2" scaleC="1" />

    <channel name="r_XQ7_thermalok" wire="digital_in_1" />

    <channel name="c_XQ7_remote" wire="digital_out_2" />

  </channels>

</board>

###
```

The above example shows an M10. The main node is a <board> with attributes that specifies it as an M10 named XQ7\_ctrl at address 7. Channels names are arbitrary, but must be unique. There are 4 channels defined, and each has a specified M10 wire. Channel "r\_XQ7\_current\_ctrl" is connected to M10 wire "analog\_in\_1". This wire correspond to the physical M10 ADC channel 1. The data type of the wire is embedded in the first part of the wire name. The direction is also embedded in the wire name, after the data type. A table of supported data types and directions is listed below. The "xxx" part of the wire name is specific to the device that the wire is assigned to.

Wire name	Data Type	User Access
<b>analog_in_xxx</b>	Double precision float (64 bit)	Read only
<b>analog_out_xxx</b>	Double precision float (64 bit)	Read/Write
<b>int_in_xxx</b>	Integer (32 bit)	Read only
<b>int_out_xxx</b>	Integer (32 bit)	Read/Write
<b>digital_in_xxx</b>	Integer (32 bit) 0=false, non-zero=true	Read
<b>digital_out_xxx</b>	Integer (32 bit) 0=false, non-zero=true	Read/Write
<b>variant_in_xxx</b>	Array of Double precision floats (64 bit)	Read

<b>variant_out_xxx</b>	Array of Double precision floats (64 bit)	Read/Write
<b>analog[n]_in_xxx</b>	Array of Double precision floats (64 bit) of length n	Read
<b>analog[n]_out_xxx</b>	Array of Double precision floats (64 bit) of length n	Read/Write
<b>string_in_xxx</b> (not supported with IG2/EPICS)	Null-terminated string	Read
<b>string_out_xxx</b> (not supported with IG2/EPICS)	Null-terminated string	Read/Write

Table 1 - Supported Data Types and User Access

## 4.1 Channel Scaling

Channels have optional scale factors to convert between user units and device units. These scale factors apply only to analog data types. This can be seen in the above M10 channel example. The channel “r\_XQ7\_current\_ctrl” specifies 2 scale factor attributes – “scaleB” and “scaleC”. These correspond to a linear scaling with the following relationship.

**<!-- y = Bx + C, where y=user units, and x=device units -->**

These scale factors are optional. If not specified in the channel node, scaleB=1 and scaleC=0.

## 4.2 Channel Limits

Channels have optional upper and lower limits. These limits apply only to analog output types, and are in user units. This can be seen in the above M10 channel example. The channel “c\_XQ7\_current\_ctrl” specifies 2 limit attributes – “limitLow” and “limitHigh”. This channel cannot be set to a value lower than limitLow or higher than limitHigh. These limits are set independently of the physical limits on the device. For a description of each physical Pyramid device, see the device data sheet and manual at <http://ptcusa.com/products.html>.

## 4.3 Channel Buffering

EPICS buffering is limited, and high-rate updates due to the processing of burst device data can cause data loss. IG2 features additional buffering controls designed to control flow of this type of data into the EPICS layer. The full design and advanced options are shown in Appendix 1.

To enable sample buffering on an individual channel, modify the channel node in the XML configuration file to contain the optional aMax parameter:

```
<channel name="bufferedchn" wire="analog_in_1" aMax="50" />
```

Set aMax to the maximum size of the device’s buffered acquisition to ensure that data is not lost. It can likely be set lower than this, as the throughput of an individual channel depends on several factors including client performance and overall number of channels in the system. aMax is an optional parameter and defaults to 0 (unbuffered).

## 4.4 Monitor Only Change

An EPICS monitor event is called on the client when a channel is updated. This can be triggered at a very high rate, even on channels that do not have constantly changing values. To disable the event unless the new value is different, modify the channel node in the XML configuration file to contain the optional monitorOnlyChange parameter:

```
<channel name="digitalchn" wire="digital_in_1" monitorOnlyChange="true" />
```

Set `monitorOnlyChange` to true if you only want to see a monitor event when the value changes. The default is set to false. This can only be used on digital, integer, or analog channels.

Advanced options are shown in Appendix 1.

## 4.5 Realtime Processor Configuration File

Some devices can be fed an xml file that will create calculations and fault condition in the realtime processor. If you have one of these files, you will need your `system.xml` file to point to it. This is done with the attribute `rtpfile="yourfile.xml"`. For example:

```
<loopcontroller type="A560" name="A560_1" ip="192.168.100.123" rtpfile="RTConfig.xml" >
```

Reading out data from the RTP file is explained in section 11.3.

## 5 Pyramid Devices

### 5.1 Supported Device Types and Wires

The system.xml hardware configuration is separated into 2 main sections, or nodes. The first is <loopcontrollers>. This section contains a complete listing of standard Pyramid hardware devices in the system. The other section is called <interpreter>. This section contains a listing of custom hardware and virtual devices.

The following devices are supported and can be found sorted in section alphabetically:

- [A560 \(see page 11\)](#)
- [B10 \(see page 12\)](#)
- [C400 \(see page 13\)](#)
- [F100 \(see page 14\)](#)
- [F3200E \(see page 15\)](#)
- [F460 \(see page 16\)](#)
- [H10 \(see page 20\)](#)
- [H20 \(see page 20\)](#)
- [I128 \(see page 22\)](#)
- [I200 \(see page 27\)](#)
- [I3200 \(see page 29\)](#)
- [I400 \(see page 29\)](#)
- [IC101 \(see page 30\)](#)
- [M10 \(see page 32\)](#)
- [M40 \(see page 32\)](#)
- [N2400 \(see page 33\)](#)

### 5.2 Overview

( <loopcontroller> section)

Each device in this section is supported in some device hierarchy. Loop controller devices can be directly communicated with over Ethernet. Fiber slave devices must be paired with a compatible Pyramid loop controller. Pyramid loop controllers can have one of two communication protocols.

G1 loop controllers = A500 (Does not support simultaneous communication with IG2 and Diagnostic)

G2 loop controllers = A560, A360, F460, I128

### 5.3 A560

```
<loopcontroller type="A560" name="???" ip="???" rtpfile="???" >
```

(G2 loop controller, direct support)

Wire	Description
<b>digital_out_initiate</b>	Initiate command (1 = Initiate, 0 = Abort)
<b>digital_out_interlock_command</b>	Set hard interlock relay state (1 = Closed, 0 = Open)
<b>digital_out_enabled_command</b>	Set soft interlock relay state (1 = Closed, 0 = Open)

<b>int_in_interlocks</b>	Bit encoded readback of interlock status (Least significant bits first) // High when CPLD is in fault state fault:1; enable_in:1; //Reflects the state of the soft relay, as determined by the commands sent enabled:1; // High if the CPLD has failed cpld_stat_A:1; // High if the CPLD has failed cpld_stat_B:1; // Reflects the key switch state (high for Diag) keyswitch:1; // command sent for diag mode diag_mode:1; // High when hard relay is closed interlock_status:1; // Reflects the intended state of the relay, as determined by the commands sent interlock_command:1; // High when the CPLD should be in the initial state initial_state:1; reserved3:4; relay_command:1; reserved4:7;
<b>digital_out_clear_errors</b>	Clear Device Errors
<b>analog[256]_in_timeslice_data</b>	Timeslice Database values configured in xml Max 256 values Convert floats to specified data types Initialized to 0 if < 256 Database values
<b>string_in_fpga</b>	FPGA Version
<b>string_in_serial_num</b>	Serial Number
<b>string_in_software_rev</b>	Software Revision
<b>string_in_secondary_fpga</b>	Secondary FPGA Version
<b>string_in_rtp_rev</b>	RTP Revision
<b>string_in_hardware_rev</b>	Hardware Revision
<b>analog_in_pos_#</b>	For use in XML configuration files in the Scan/Dose system, this allows the a560 to read single analog values sent from each slave I128.

## 5.4 B10

<board type="B10" ... > (Fiber slave device, G1 loop controller support only)

Wire	Description
<b>digital_in_1</b>	TTL digital input channel 1
<b>digital_in_2</b>	TTL digital input channel 2
<b>digital_in_3</b>	TTL digital input channel 3
<b>digital_in_4</b>	TTL digital input channel 4
<b>digital_in_5</b>	TTL digital input channel 5

<b>digital_in_6</b>	TTL digital input channel 6
<b>digital_in_7</b>	TTL digital input channel 7
<b>digital_in_8</b>	TTL digital input channel 8
<b>digital_out_1</b>	TTL digital output channel 1
<b>digital_out_2</b>	TTL digital output channel 2
<b>digital_out_3</b>	TTL digital output channel 3
<b>digital_out_4</b>	TTL digital output channel 4
<b>digital_out_5</b>	TTL digital output channel 5
<b>digital_out_6</b>	TTL digital output channel 6
<b>digital_out_7</b>	TTL digital output channel 7
<b>digital_out_8</b>	TTL digital output channel 8

## 5.5 C400

<board type="C400" ... > (G2 loop controller, direct support)

Wire	Description
<b>digital_in_running</b>	Running state (1=running, 0=not running)
<b>digital_in_paused</b>	Paused state (1=paused, 0=not paused)
<b>digital_in_stopped</b>	Stopped state (1=stopped, 0=not stopped)
<b>int_in_counts_n</b>	Counts for the 4 channels (n=1-4)
<b>analog_in_rate_n</b>	Rate for the 4 channels (n=1-4)
<b>analog_in_bias_n</b>	Voltage readback for the 4 channels (n=1-4)
<b>digital_out_initiate</b>	Initiate/Abort acquisition control command (1=initiate, 0=abort)
<b>digital_out_polarity_n</b>	Discriminator polarities for the 4 channels (n=1-4) (1=positive,0=negative)
<b>digital_out_pulse_enable_n</b>	Pulse control for the 4 channels (n=1-4) (1=enable,0=disable)
<b>int_out_accum_mode</b>	Accumulate mode (0-1) (C400_SingleIntegrations = 0, C400_AccumulatedIntegrations = 1)
<b>int_out_trig_buf</b>	Data buffer size
<b>int_out_trig_bur</b>	Burst size
<b>int_out_trig_mode</b>	Trigger mode (0-6) (C400_CustomAcquisitionMode = 0, C400_InternalAcquisitionMode = 1, C400_ExternalStartAcquisitionMode = 2, C400_ExternalStartStopAcquisitionMode = 3, C400_ExternalStartHoldAcquisitionMode = 4, C400_ExternalWindowedAcquisitionMode = 5, C400_DiscriminatorSweepMode = 6)

<b>int_out_trig_source_start</b>	Start trigger source (0-3) (InternalTriggerSource = 0, BNCTriggerSource = 1, OpticalTriggerSource = 2, BNC_and_OpticalTriggerSource = 3)
<b>int_out_trig_source_stop</b>	Stop trigger source (0-3) (InternalTriggerSource = 0, BNCTriggerSource = 1, OpticalTriggerSource = 2, BNC_and_OpticalTriggerSource = 3)
<b>int_out_trig_source_pause</b>	Pause trigger source (0-3) (InternalTriggerSource = 0, BNCTriggerSource = 1, OpticalTriggerSource = 2, BNC_and_OpticalTriggerSource = 3)
<b>int_out_pulse_period</b>	Pulse period (nsec)
<b>int_out_pulse_width</b>	Pulse width (nsec)
<b>analog_out_low_limit_n</b>	Discriminator low level (volts)
<b>analog_out_high_limit_n</b>	Discriminator high level (volts)
<b>analog_out_bias_n</b>	Voltage control for the 4 channels (volts)
<b>analog_out_period</b>	Integration period (sec)

## 5.6 F100

<board type="F100" ... > (Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
<b>analog_in_1</b>	Current input (A)
<b>digital_out_initiate</b>	Initiate measurement (1=initiate, 0 =abort)
<b>int_out_range</b>	Range select (0-16) Sixteen total, thirteen unique ( 0=auto 1 = 1 $\mu$ A 2 = 2 $\mu$ A 3 = 5 $\mu$ A 4 = 10 $\mu$ A 5 = 10 $\mu$ A 6 = 20 $\mu$ A 7 = 50 $\mu$ A 8 = 100 $\mu$ A 9 = 100 $\mu$ A 10 = 200 $\mu$ A 11 = 500 $\mu$ A 12 = 1 mA 13 = 1 mA 14 = 2 mA 15 = 5 mA 16 = 10 mA )
<b>int_out_hv_bias</b>	High voltage control (V)
<b>analog_in_hv_bias</b>	High voltage readback (V) Updates every 50ms

<b>digital_out_calibration_current</b>	Calibration Current 1 = On (5/500 uA) 0 = Off
<b>digital_out_actuator</b>	Actuator 1 = On 0 = Off
<b>analog_out_averaging_period</b>	Averaging Period (s) 1e-4 to 1e0

## 5.7 F3200E

<board type="F3200E" ... > (G2 loopcontroller support)

Wire	Description
<b>variant_in_data</b>	32 elements of channel data (A) 1 element of sample info A (bit convert to int) 1 element of sample info B (bit convert to int) 4 elements of ADC 3 elements of DAC command 1 overrange bitmask
<b>analog_out_adc_1 (not yet added)</b>	ADC out channel 1 (V)
<b>analog_out_adc_2 (not yet added)</b>	ADC out channel 2 (V)
<b>analog_out_adc_3 (not yet added)</b>	ADC out channel 3 (V)
<b>analog_in_adc_1 (not yet added)</b>	ADC in channel 1 (V)
<b>analog_in_adc_2 (not yet added)</b>	ADC in channel 2 (V)
<b>analog_in_adc_3 (not yet added)</b>	ADC in channel 3 (V)
<b>analog_in_adc_4 (not yet added)</b>	ADC in channel 4 (V)
<b>digital_in_1 (not yet added)</b>	TTL digital input channel 1
<b>digital_in_2 (not yet added)</b>	TTL digital input channel 2
<b>digital_in_3 (not yet added)</b>	TTL digital input channel 3
<b>digital_in_4 (not yet added)</b>	TTL digital input channel 4
<b>digital_out_1 (not yet added)</b>	TTL digital output channel 1
<b>digital_out_2 (not yet added)</b>	TTL digital output channel 2
<b>digital_out_3 (not yet added)</b>	TTL digital output channel 3
<b>digital_out_4 (not yet added)</b>	TTL digital output channel 4
<b>digital_out_initiate</b>	Initiate measurement (1=initiate, 0 =abort)
<b>variant_in_range</b>	16 elements of range (F3200_10uA Range = 0 F3200_100uA Range = 1 F3200_1mA Range = 2 F3200_100mA Range = 3)

<b>variant_out_range_set</b>	1 element of bank (1-16) If bank == -1, all ranges are set 1 element of range (0-3) (F3200_10uA Range = 0 F3200_100uA Range = 1 F3200_1mA Range = 2 F3200_100mA Range = 3)
<b>digital_out_buffered_acquisition</b>	Buffer Contiguous Data (1 = Enable, 0 = Disable)
<b>int_out_acquisition_mode</b>	Acquisition mode (0-7) (F3200_CustomAcquisitionMode = 0, F3200_InternalAcquisitionMode = 1, F3200_ExternalStartAcquisitionMode = 2, F3200_ExternalStartStopAcquisitionMode = 3, F3200_ExternalStartHoldAcquisitionMode = 4, F3200_ExternalWindowedAcquisitionMode = 5, F3200_SweepMode = 6 F3200_EncoderDrivenMode = 7)
<b>int_out_adc_rate</b>	ADC rate (Hz)
<b>int_out_conversions_per_sample</b>	Conversions per sample
<b>int_out_start_trigger_source</b>	Start trigger source (0-3) (InternalTriggerSource = 0, BNCTriggerSource = 1, OpticalTriggerSource, BNC_and_OpticalTriggerSource)
<b>int_out_pause_trigger_source</b>	Start trigger source (0-3) (InternalTriggerSource = 0, BNCTriggerSource = 1, OpticalTriggerSource, BNC_and_OpticalTriggerSource)
<b>int_out_stop_trigger_source</b>	Start trigger source (0-3) (InternalTriggerSource = 0, BNCTriggerSource = 1, OpticalTriggerSource, BNC_and_OpticalTriggerSource)
<b>int_out_stop_count</b>	Stop Count
<b>int_out_burst_count</b>	Burst Size (0 = unlimited)
<b>int_out_register_offset</b>	Offset from base_address to read or write
<b>int_out_register_contents</b>	Contents read from address, or contents to write to address
<b>int_out_base_address</b>	Base address (initialized as 0x08810000)
<b>digital_out_register_command</b>	Get register at address or Set register address with contents (1 = set, 0 = get)

## 5.8 F460

<loopcontroller type="F460" ... > (G2 loop controller, direct support)

Wire	Description
<b>analog_in_1</b>	ADC channel 1 (volts)
<b>analog_in_2</b>	ADC channel 2 (volts)

<b>analog_in_current_1</b>	Current channel 1 (A)
<b>analog_in_current_2</b>	Current channel 2 (A)
<b>analog_in_current_3</b>	Current channel 3 (A)
<b>analog_in_current_4</b>	Current channel 4 (A)
<b>analog_in_channel_1</b>	Current channel 1 w/sensor compensations (A)
<b>analog_in_channel_2</b>	Current channel 2 w/sensor compensations (A)
<b>analog_in_channel_3</b>	Current channel 3 w/sensor compensations (A)
<b>analog_in_channel_4</b>	Current channel 4 w/sensor compensations (A)
<b>analog_in_x_pos</b>	Calculated x position
<b>analog_in_y_pos</b>	Calculated y position
<b>analog_in_bias</b>	Voltage
<b>int_in_max_bias</b>	Max bias readout
<b>analog_in_process_value</b>	Process value (servo)
<b>analog_in_process_target</b>	Process target (servo)
<b>variant_in_data</b>	17 elements in total: 4 elements of current data 4 elements of channel data 1 element of xpos 1 of ypos 2 elements of analog in data 4 elements of analog out data 1 element of high voltage
<b>analog[4081]_in_buffered_variant_data</b>	Contains 240 variant_in_data structures (240*17=4080). Flatted, so the first 17 values are from sample 1, the next 17 from sample 2, etc. Plus one element at the beginning containing the number of valid samples in the array (0-240).
<b>digital_out_initiate</b>	Initiate measurement (1=initiate, 0 =abort)
<b>int_out_range_1</b>	Channel 1 range (0-3) (F460_1uA_Range = 0, F460_10uA_Range = 1, F460_100uA_Range = 2, F460_1mA_Range = 3)
<b>int_out_range_2</b>	Channel 2 range (same as channel 1)
<b>int_out_range_3</b>	Channel 3 range (same as channel 1)
<b>int_out_range_4</b>	Channel 4 range (same as channel 1)
<b>int_out_monitor</b>	Monitor out mode (0-3) (F460_Current = 0, F460_Sensor = 1, F460_Position = 2, F460_ManualOutput = 3)
<b>analog_out_1</b>	Analog out channel 1 (V)
<b>analog_out_2</b>	Analog out channel 2 (V)

<b>analog_out_3</b>	Analog out channel 3 (V)
<b>analog_out_4</b>	Analog out channel 4 (V)
<b>analog_out_bias</b>	External bias (V)
<b>analog_out_integration_time</b>	Integration time (seconds)
<b>analog_out_dac_low_limit</b>	DAC low limit (servo)
<b>analog_out_dac_high_limit</b>	DAC high limit (servo)
<b>analog_out_kp</b>	Kp PID parameter (servo)
<b>analog_out_ki</b>	Ki PID parameter (servo)
<b>analog_out_low_current_limit</b>	Low current limit (servo)
<b>analog_out_reference</b>	Reference (servo)
<b>int_out_servo_period</b>	Servo period (microseconds)
<b>int_out_servo_mode</b>	Servo mode (1-7, see device manual)
<b>digital_out_servo_enable</b>	Servo enable (1=enable, 0=disable)
<b>int_out_register_offset</b>	Offset from base_address to read or write
<b>int_out_register_contents</b>	Contents read from address, or contents to write to address
<b>digital_out_register_command</b>	Get register at address or Set register address with contents (1 = set, 0 = get)
<b>int_out_base_address</b>	Base address (initialized as 0x08810000)
<b>string_in_firmware</b>	Firmware Version
<b>string_in_fpga</b>	FPGA Version
<b>string_in_serial_num</b>	Serial Number
<b>string_in_software_rev</b>	Software Revision
<b>string_in_secondary_fpga</b>	Secondary FPGA Version
<b>string_in_rtp_rev</b>	RTP Revision
<b>string_in_hardware_rev</b>	Hardware Revision
<b>int_out_position_calculation</b>	(1 = Split Calculation, 0 = Quadrant Calculation )
<b>digital_out_buffered_acquisition</b>	Buffer Contiguous Data (1 = Enable, 0 = Disable)

<b>int_out_acquisition_mode</b>	Acquisition mode (0-5) (F460_CustomAcquisitionMode = 0, F460_InternalAcquisitionMode = 1, F460_ExternalStartAcquisitionMode = 2, F460_ExternalStartStopAcquisitionMode = 3, F460_ExternalStartHoldAcquisitionMode = 4, F460_ExternalWindowedAcquisitionMode = 5)
<b>int_out_start_trigger_source</b>	Start trigger source (0-1) (InternalTriggerSource = 0, BNCTriggerSource = 1)
<b>int_out_pause_trigger_source</b>	Start trigger source (0-1) (InternalTriggerSource = 0, BNCTriggerSource = 1)
<b>int_out_stop_trigger_source</b>	Start trigger source (0-1) (InternalTriggerSource = 0, BNCTriggerSource = 1)
<b>int_out_bnc_start_gate</b>	Trigger gate polarity (1 = Falling edge gate, 0 = Rising edge gate)
<b>int_out_stop_count</b>	Stop Count
<b>int_out_burst_count</b>	Burst Size
<b>int_out_calibration_source</b>	Calibration Source (0-2) (No Calibration Source = 0, Internal Low = 1, Internal High = 2)
<b>int_out_calibration_channel</b>	Channel to receive internal calibration source (0-3)
<b>int_out_calibration_command</b>	Calibrate Channel 0-3: F460 Channel -1: All Channels
<b>digital_out_clear_calibrations</b>	Clear Calibrations
<b>variant_out_hvdac_calibration</b>	3-element array [valid, gain, offset]
<b>variant_out_hvadc_calibration</b>	3-element array [valid, gain, offset]
<b>variant_out_calibration_range_1</b>	Calibration for range 1uA 9-element array: [valid, gain[0], offset[0], ..., gain[3], offset[3]]
<b>variant_out_calibration_range_2</b>	Calibration for range 10uA (Same as range 1)
<b>variant_out_calibration_range_3</b>	Calibration for range 100uA (Same as range 1)
<b>variant_out_calibration_range_4</b>	Calibration for range 1mA (Same as range 1)
<b>variant_out_analoginput_calibration</b>	5-element array [valid, gain[0], offset[0], gain[1], offset[1]]
<b>variant_out_analogoutput_calibration</b>	9-element array [valid, gain[0], offset[0], ..., gain[3], offset[3]]

## 5.9 H10

<board type="H10" ... > (Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
analog_in_1	Old probe: dBdT, New probe: voltage
analog_in_2	Field

## 5.10 H20

<board type="H20" ... > (Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
digital_out_initiate	Initiate command (1 = Initiate, 0 = Abort)
analog_in_probe_field_1	Field measured by probe 1 (G)
analog_in_probe_field_2	Field measured by probe 2 (G)
analog_in_probe_temperature_1	Temperature measured by probe 1 (C)
analog_in_probe_temperature_2	Temperature measured by probe 2 (C)
analog_in_adc_1	ADC1 measurement (V)
analog_in_adc_2	ADC2 measurement (V)
analog_out_dac_1	DAC (V) (Manual Mode Only)
analog_out_dac_2	DAC (V) (Manual Mode Only)
int_out_range_1	Probe 1 range 0: 1x 1: 4x 2: 10x 3: 40x
int_out_range_2	Probe 2 range Same mapping as int_out_range_1
int_out_mode_1	Probe 1 mode Mode_Manual = 0x0, Mode_Monitor = 0x1, Mode_DigitalClosedLoop = 0x2, Mode_AnalogClosedLoop = 0x3, Mode_FastMonitor = 0x4
int_out_mode_2	Probe 1 mode Same mapping as mode_out_1
analog_out_setpoint_1	Field Setpoint (G) (Digital Closed Loop Mode Only)
analog_out_setpoint_2	Field Setpoint (G) (Digital Closed Loop Mode Only)
analog_out_averaging_period	Averaging period (s)
int_out_stop_count	If < 1, no stop

<b>string_in_serial_num_1</b>	Probe 1 serial number
<b>string_in_serial_num_2</b>	Probe 2 serial number
<b>analog_in_calib_temp_1</b>	Calibration Temperature
<b>analog_in_calib_temp_2</b>	Calibration Temperature
<b>analog_out_proportional_1</b>	Proportional (Kp)
<b>analog_out_proportional_2</b>	Proportional (Kp)
<b>analog_out_outmax_1</b>	Out Max (V)
<b>analog_out_outmax_2</b>	Out Max (V)
<b>digital_out_positive_output_1</b>	Positive only output ( 0 = Positive only disabled 1 = Positive only enabled )
<b>digital_out_positive_output_2</b>	Positive only output ( 0 = Positive only disabled 1 = Positive only enabled )
<b>analog_out_slew_limit_1</b>	Slew Limit (V/s)
<b>analog_out_slew_limit_2</b>	Slew Limit (V/s)
<b>analog_out_setpoint_gain_1</b>	Setpoint Gain (G/V) (Analog Closed Loop Mode Only)
<b>analog_out_setpoint_gain_2</b>	Setpoint Gain (G/V) (Analog Closed Loop Mode Only)
<b>string_in_h2o_serial_num</b>	H2O Serial Number
<b>analog[2]_out_h2o_A_field_1x</b>	H2O Field 1x Calibration Side A [gain, offset] [0]: Gain [1]: Offset
<b>analog[2]_out_h2o_B_field_1x</b>	H2O Field 1x Calibration Side B [0]: Gain [1]: Offset
<b>analog[2]_out_h2o_A_field_10x</b>	H2O Field 10x Calibration Side A [0]: Gain [1]: Offset
<b>analog[2]_out_h2o_B_field_10x</b>	H2O Field 10x Calibration Side B [0]: Gain [1]: Offset
<b>analog[2]_out_h2o_A_temp</b>	Temp ADC Side A [0]: Gain [1]: Offset
<b>analog[2]_out_h2o_B_temp</b>	Temp ADC Side B [0]: Gain [1]: Offset
<b>analog[2]_out_h2o_A_analoginput</b>	H2O Analog Input Side A [0]: Gain [1]: Offset
<b>analog[2]_out_h2o_B_analoginput</b>	H2O Analog Input Side B [0]: Gain [1]: Offset

<b>analog[2]_out_h20_A_analogoutput</b>	H20 Analog Output Side A [0]: Gain [1]: Offset
<b>analog[2]_out_h20_B_analogoutput</b>	H20 Analog Output Side B [0]: Gain [1]: Offset
<b>analog[2]_out_probe_A_field_1x</b>	Probe A Field 1x Calibration [0]: Gain [1]: Offset
<b>analog[2]_out_probe_B_field_1x</b>	Probe B Field 1x Calibration [0]: Gain [1]: Offset
<b>analog[2]_out_probe_A_field_4x</b>	Probe A Field 4x Calibration [0]: Gain [1]: Offset
<b>analog[2]_out_probe_B_field_4x</b>	Probe B Field 4x Calibration [0]: Gain [1]: Offset

### 5.11 I128

<loopcontroller type="I128" ... > (G2 loop controller, direct support)

Wire	Description
<b>analog[135]_in_current</b>	128 elements of channel data (A) -20000 = unable to convert to A 1 element of HCC 1 element of seconds (timestamp) 1 element of nanoseconds 4 elements of over-range/under-range bits
analog_in_fit1_position	beam position, calculated from Gaussian fit of 128 strips of strip chamber
analog_in_fit1_width	beam sigma, calculated from Gaussian fit of 128 strips of strip chamber
analog_in_fit1_amplitude	calculated from Gaussian fit of 128 strips of strip chamber
<b>int_out_curve_fit_chamber_type</b>	0: Strip Chamber 1: PX2 Chamber 2: PX3 Chamber
<b>analog_out_discriminator</b>	Discriminator for pixelated data

<b>analog[163]_in_px_data</b>	<p>163-element array of PX data  Pixelated data is enabled by the curve_fit_chamber_type wire:  [0...143] Pixel data (A)  -20000 = unable to convert to A  [144] HCC  [145] Seconds (timestamp)  [146] Nanoseconds  [147] Valid  Invalidated by any over-range/under-range channels  [148] Rotation (degrees)  [149] Mean X (mm)  [150] Mean Y (mm)  [151] Sigma X (mm)  [152] Sigma Y (mm)  [153] Elongation  [154] Integral (A)  [155] Major (mm)  [156] Minor (mm)  [157] Max (A)  [158...162] Over-range/under-range bits</p>
<b>digital_in_connection_status</b>	<p>Status of connection between I128 and IG2  1: Connected  0: Disconnected</p>
<b>digital_in_measuring</b>	Measuring readback
<b>analog_in_external_hv</b>	High voltage readback (volts)
<b>analog_out_hv</b>	High voltage command (volts)
<b>digital_out_enable_external_hv</b>	High voltage output (1 = enable, 0 = disable)
<b>analog_in_hcc</b>	High current channel readback (A)
<b>analog_in_range_current</b>	Range (A)
<b>analog_in_range_charge</b>	Range (C)
<b>analog_out_integration_time</b>	Integration time command (s)
<b>int_in_integration_time</b>	Integration time readback (ns)
<b>int_out_conversions_per_sample</b>	Conversions per sample (1-255)
<b>int_out_range_hcc</b>	<p>High current channel range  I128HCC_1uA_Range = 0x0,  I128HCC_5uA_Range = 0x1,  I128HCC_10uA_Range = 0x2,  I128HCC_20uA_Range = 0x3,</p>
<b>digital_out_initiate</b>	<p>Initiate command  (1 = Initiate, 0 = Abort)</p>
<b>int_out_ion_chamber_mode</b>	<p>Ion chamber mode  ( 0 = Standard Mode  1 = IC Mode)</p>
<b>analog_in_ic_temp</b>	Ion Chamber Temperature (I)
<b>analog_in_ic_pressure</b>	Ion Chamber Pressure (PSI)
<b>analog_in_ic_humidity</b>	Ion Chamber Humidity (%RH)

<b>analog_in_ic_reference</b>	Ion Chamber Reference (V)
<b>analog_out_dac_1</b>	DAC out 1
<b>analog_out_dac_2</b>	DAC out 2
<b>analog_in_adc_1</b>	ADC In 1
<b>analog_in_adc_2</b>	ADC In 2
<b>analog_out_averaging_period</b>	Averaging Period ( 25 = 1e-4 250 = 1e-3 2500 = 1e-2 25000 = 1e-1 41675 = 1.667e-1 250000 = 1.0 )
<b>analog[258]_out_offset_vector</b>	258-element array of gain/offset pairs. [0] Gain on Channel 0 [1] Offset on Channel 0 .... [254] Gain on Channel 127 [255] Offset on Channel 127 [256] HCC Gain [257] HCC Offset Setting this wire immediately sets the background offset vector.
<b>digital_out_clear_offset_vector</b>	Clears the background offset vector (gains = 0, offsets = 1)
<b>string_in_firmware</b>	Firmware Version
<b>string_in_fpga</b>	FPGA Version
<b>string_in_serial_num</b>	Serial Number
<b>string_in_software_rev</b>	Software Revision
<b>string_in_secondary_fpga</b>	Secondary FPGA Version
<b>string_in_rtp_rev</b>	RTP Revision
<b>string_in_hardware_rev</b>	Hardware Revision
<b>string_in_ip_address</b>	IP Address of device
<b>int_out_filter</b>	HCC Filter
<b>digital_out_combine_channels</b>	HCC Combine Channels Enable
<b>analog_out_monitor_charge</b>	HCC Monitor Charge
<b>digital_out_align_channel_data</b>	HCC Align with Channel Data Enable
<b>analog_out_target_dose</b>	Target Charge (nC) (This and next wire should be tested thoroughly)
<b>digital_out_opt_enable</b>	“Enable Beam” fiberoptic output
<b>digital_in_opt_enabled</b>	“Enable Beam” fiberoptic ( 1 = enabled, 0 = disabled)

<b>int_out_digitals</b>	Bit encoded digital outputs (0-15) bit0 = Digital 1 bit1 = Digital 2 bit2 = Digital 3 bit3 = Digital 4
<b>int_in_digitals</b>	Bit encoded digital inputs (0-15) bit0 = Digital 1 bit1 = Digital 2 bit2 = Digital 3 bit3 = Digital 4
<b>analog_in_hcc_processed</b>	HCC data in Amps. May be processed (filtered)
<b>digital_in_hcc_target_reached</b>	(1 = target reached)
<b>analog_in_hcc_dose</b>	FPGA2 v1.4.10 and later: Dose accumulated on the HCC in coulombs. FPGA2 v1.4.9 and earlier: HCC dose remaining in coulombs.
<b>int_out_start_trigger_source</b>	Start trigger source (0-1) (InternalTriggerSource = 0, BNCTriggerSource = 1)
<b>int_out_pause_trigger_source</b>	Start trigger source (0-1) (InternalTriggerSource = 0, BNCTriggerSource = 1)
<b>int_out_stop_trigger_source</b>	Start trigger source (0-1) (InternalTriggerSource = 0, BNCTriggerSource = 1)
<b>int_out_bnc_start_gate</b>	Trigger gate polarity (1 = Falling edge gate, 0 = Rising edge gate)
<b>int_out_stop_count</b>	Stop Count (0 = unlimited)
<b>int_out_burst_count</b>	Burst Size (0 = unlimited)
<b>int_out_register_offset</b>	Offset from base_address to read or write
<b>int_out_register_contents</b>	Contents read from address, or contents to write to address
<b>digital_out_register_command</b>	Get register at address or Set register address with contents (1 = set, 0 = get)
<b>int_out_base_address</b>	Base address (initialized as 0x08810000)

<b>int_in_interlock_readbacks</b>	<p>Bit encoded readback of interlock status (Least significant bits first)</p> <pre> cmd:8; reserved1:1; reserved2:1; // High when CPLD is in fault state fault:1; enable_in:1; //Reflects the state of the Enabled line, as determined by the commands sent enabled:1; // High if the CPLD has failed cpld_stat_A:1; // High if the CPLD has failed cpld_stat_B:1; // Reflects the key switch state (high for Diag) keyswitch:1; // command sent for diag mode diag_mode:1; // High when relay is closed interlock_status:1; // Reflects the intended state of the relay, as determined by the commands sent interlock_command:1; // High when the CPLD should be in the initial state initial_state:1; reserved3:4; relay_command:1; reserved4:7;                     </pre>
<b>digital_out_relay_enable</b>	( 1 = enable, 0 = disable)
<b>digital_out_actuator_enable</b>	( 1 = enable, 0 = disable)
<b>digital_out_test_ab</b>	( 0 = Test A, 1 = Test B) no readback
<b>digital_out_interlock_enable</b>	( 0 = Enable, 1 = Disable) no readback
<b>digital_out_calibration_source</b>	Calibration Source I128_NoCalibrationSource = 0, I128_InternalCalibrationSource = 1,
<b>int_out_calibration_channel</b>	Calibration channel (1-129) 1-128 = strip select 129 = HCC
<b>analog[260]_out_current_calibration</b>	4 elements of valid (128 bits), followed by gain/offset pairs [valid x4, gain[0], offset[0], ..., gain[127], offset[127]]
<b>int_out_calibrate</b>	Calibrate Command -1: Calibrate all channels 0-127: Calibrate one channel 128: Calibrate HCC
<b>digital_out_clear_calibrations</b>	Clear all calibrations
<b>digital_out_buffered_acquisition</b>	Buffer Contiguous Data (1 = Enable, 0 = Disable)
<b>digital_out_clear_errors</b>	Clear Device Errors
<b>digital_out_set_interlock</b>	Set interlock On or Off

<b>string_out_configure_xml</b>	Send filename of xml file to upload Readbacks: "InputFileName" at startup "success" if successful upload "failure" if failed
<b>analog[256]_in_timeslice_data</b>	Timeslice Database values configured in xml Max 256 values Convert floats to specified data types Initialized to 0 if < 256 Database values
<b>analog_out_monitor_charge</b>	Controls the charge required for each pulse emitted out the monitor connector.
<b>analog_in_ic_monitor_charge</b>	Reads back the programmed charge required for each pulse emitted out the monitor connector.
<b>digital_out_fast_message_enable</b>	1=enable fast message mode, 0=disable
<b>string_out_fast_message_configuration</b>	Configures the destination for the fast messages. The destination IP, port, and physical MAC address are specified, comma delimited. For example, ip,port,macid ip – IP address of client as a standard ipv4 string. port – is the port number of the client, recommended > 50000 mac – physical address of the client  Example: 192.168.100.105,51000,98:90:96:E3:88:83
<b>analog_out_pos_offset</b>	For use in XML configuration files only for Scan/Dose systems, write any analog value to this wire and it will show up at the master A560.

## 5.12 I200

<board type="I200" name="TODO" address="TODO">  
(Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
<b>digital_out_initiate</b>	Initiate/abort acquisition control 0: Abort 1: Initiate
<b>analog[5]_in_data</b>	5-element array of channel data (A) [0] Trigger Count [1] Channel A current (A) [2] Channel B current (A) [3] Channel A charge (C) [4] Channel B charge (C) -10000 = overrange
<b>analog_in_channel_a</b>	Channel A current (A)
<b>analog_in_channel_b</b>	Channel B current (A)
<b>analog[5]_in_buffered_data</b>	5-element array of buffered channel data. This has same structure as <b>analog[5]_in_data</b> . This wire is used when trigger points is non-infinite. Use aMax in xml to ensure all samples are collected.

<b>int_out_trigger_source</b>	Set trigger source (0-3) 0: Internal 1: External Start 2: External Start Stop 3: External Start Hold
<b>int_out_trigger_connector</b>	Set trigger connector (1-2) 1: BNC 2: Optical
<b>int_out_trigger_points</b>	Set trigger points -1: Infinite Any other positive value starts buffered mode, which uses <b>analog[5]_in_buffered_data</b>
<b>int_in_trigger_count</b>	Trigger Count Is updated in both buffered and non-buffered mode
<b>int_out_accumulate_mode</b>	Accumulate mode (0-2) (No Compensation = 0, Estimated = 1 Lossless = 2)
<b>digital_in_actuator_a</b>	Actuator A 1 = On, 0 = Off
<b>digital_in_actuator_b</b>	Actuator B 1 = On, 0 = Off
<b>int_out_capacitor</b>	Capacitor configuration control (0-1) 0 = 10pF 1 = 1000pF
<b>analog_out_integration_period</b>	Set integration period (sec)
<b>int_out_subsamples</b>	Set # of subsamples (1-255)
<b>analog_out_range</b>	Set range (A)
<b>int_out_hv_external_bias</b>	Set high voltage (V)
<b>analog_in_hv_external_bias</b>	Read high voltage (V)
<b>int_in_external_bias_max</b>	High Voltage Max (V)
<b>int_out_cap1_reset_time</b>	10pF reset time (microseconds)
<b>int_out_cap2_reset_time</b>	1000pF reset time (microseconds)
<b>int_out_cap1_settle_time</b>	10pF settle time (microseconds)
<b>int_out_cap2_settle_time</b>	1000pF settle time (microseconds)
<b>int_out_setup_time</b>	Setup Time (microseconds)
<b>analog[4]_in_calibration_gain</b>	Calibration Gains [0]: Channel A Cap 1 Gain [1]: Channel B Cap 1 Gain [2]: Channel A Cap 2 Gain [3]: Channel B Cap 2 Gain
<b>int_out_calibration_source</b>	Calibration Current Channel (0-2) 0 = Off 1 = Channel A calibration current 2 = Channel A calibration current

<b>digital_out_calibrate</b>	Calibrate Command
<b>analog_in_sample_rate</b>	Sample Rate (Hz)

## 5.13 I3200

<board type="I3200" ... > (Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
<b>variant_in_current</b>	32-element array of channel data (A)
<b>digital_in_measuring</b>	I3200 measuring mode indicator
<b>analog_in_bias</b>	High voltage readback (volts)
<b>digital_in_actuator_limitA</b>	Actuator limit A readback
<b>digital_in_actuator_limitB</b>	Actuator limit B readback
<b>digital_out_actuator</b>	Actuator position control
<b>analog_out_bias</b>	High voltage control (volts)
<b>analog_out_period</b>	I3200 integration time configuration control
<b>int_out_capacitor</b>	I3200 capacitor configuration control (0 or 1)
<b>digital_out_initiate</b>	I3200 initiate acquisition control
<b>int_out_trigger_mode</b>	Set trigger source (1-6) 1: Internal 2: External Start 3: External Gated 4: Message 5: External Start/Stop 6: External Start/Hold
<b>int_out_trigger_start</b>	Trigger Start (1-2) 1: BNC 2: Optical
<b>int_out_trigger_points</b>	Set trigger points -1: Infinite Any other positive value starts buffered mode (Buffer Contiguous Data). Use aMax in xml to ensure all buffered samples are collected.

## 5.14 I400

<board type="I400" name="TODO" address="TODO">  
(Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
<b>digital_out_initiate</b>	Initiate/abort acquisition control 0: Abort 1: Initiate

<b>analog[9]_in_data</b>	9-element array of channel data (A) [0] Trigger Count [1] Channel A current (A) [2] Channel B current (A) [3] Channel C current (A) [4] Channel D current (A) [5] Channel A charge (C) [6] Channel B charge (C) [7] Channel C charge (C) [8] Channel D charge (C) -10000 = overrange
<b>analog_in_channel_a</b>	Channel A current (A)
<b>analog_in_channel_b</b>	Channel B current (A)
<b>analog_in_channel_c</b>	Channel C current (A)
<b>analog_in_channel_d</b>	Channel D current (A)
<b>int_out_trigger_source</b>	Set trigger source (1-5) 1: Internal 2: External Start 3: External Gated 4: Message 5: External Start/Stop
<b>int_out_trigger_points</b>	Set trigger points -1: Infinite Any other positive value starts buffered mode. Use aMax in xml to ensure all buffered samples are collected.
<b>int_in_trigger_count</b>	Trigger Count Is updated in both buffered and non-buffered mode
<b>int_out_capacitor</b>	Capacitor configuration control (0-1) 0: 10pF 1: 1000pF
<b>analog_out_integration_period</b>	Set integration period (sec)
<b>int_out_subsamples</b>	Set # of subsamples (1-255)
<b>analog_out_range</b>	Set range (A)
<b>int_out_hv_external_bias</b>	Set high voltage (V)
<b>analog_in_hv_external_bias</b>	Read high voltage (V)
<b>int_out_signal_bias</b>	Set Signal Bias (V)
<b>analog_in_signal_bias</b>	Signal Bias Readback (V)
<b>analog_in_temperature</b>	Temperature Readback (C)
<b>analog_in_pressure</b>	Pressure Readback (Pa)

## 5.15 IC101

<board type="IC101" ... > (Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
<b>analog_in_current</b>	Current input (A)
<b>analog_in_accumulated_charge</b>	Accumulated charge (C)

<b>analog_out_range</b>	Range select (A) and Readback
<b>digital_out_initiate</b>	Initiate command
<b>digital_out_accumulate_charge</b>	Accumulate command (1=start accumulation, 0=stop accumulation)
<b>analog_out_integration_period</b>	Set Integration Period (1e-4 to 1e0)
<b>int_out_resolution</b>	Set Resolution (16-20)
<b>int_out_capacitor</b>	Set Capacitor 0: small (100 pF by default) 1: large (3300 pF by default)
<b>digital_out_calibrate</b>	Calibrate
<b>digital_out_save_calibration</b>	Saves active calibration
<b>digital_out_recall_calibration</b>	Recalls stored calibration
<b>digital_out_calibration_source</b>	Calibration Source (On/Off)
<b>int_out_external_bias</b>	Set external Bias in volts If fitted; 0 to max or 0 to -max
<b>int_out_external_bias_max</b>	Set Max External Bias in volts
<b>int_out_trigger_mode</b>	Trigger Source 0: Internal 1: External
<b>int_in_trigger_count</b>	Trigger Count
<b>int_out_frequency_monitor_output_mode</b>	Frequency Monitor Output Mode (1-3) 1: Absolute 2: Positive 3: Negative
<b>int_out_frequency_monitor_mapping</b>	Frequency Monitor Mapping (0-4) 0: 1MHz 1: 100kHz 2: 10kHz 3: 1kHz 4: 100Hz
<b>digital_out_monitor_log_mode</b>	Monitor Log Mode 0: Off 1: On
<b>analog_out_monitor_scale</b>	Monitor Scale
<b>analog[5]_in_calibration</b>	Calibration Factors 0: Valid bit 1: Gain for small capacitor 2: Offset for small capacitor 3: Gain for big capacitor 4: Offset for big capacitor
<b>int_in_status_byte</b>	Read digitals bit0 = measuring bit1 = waiting trigger bit2 = calibrated bit3 = HV enabled bit4 = external gate present bit5 = new data

<b>string_in_serial_num</b>	Serial Number
-----------------------------	---------------

## 5.16 M10

<board type="M10" ... > (Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
<b>digital_out_initiate</b>	Initiate command (1 = Initiate, 0 = Abort)
<b>int_out_stop_count</b>	If < 1, no stop
<b>analog_in_1</b>	ADC channel 1 (volts)
<b>analog_in_2</b>	ADC channel 2 (volts)
<b>analog_out_1</b>	DAC channel 1 (volts)
<b>analog_out_2</b>	DAC channel 2 (volts)
<b>digital_in_1</b>	TTL digital input channel 1
<b>digital_in_2</b>	TTL digital input channel 2
<b>digital_in_3</b>	TTL digital input channel 3
<b>digital_in_4</b>	TTL digital input channel 4
<b>digital_out_1</b>	TTL digital output channel 1
<b>digital_out_2</b>	TTL digital output channel 2
<b>digital_out_3</b>	TTL digital output channel 3
<b>digital_out_4</b>	TTL digital output channel 4

## 5.17 M40

<board type="M40" ... > (Fiber slave device, G1, G2 loopcontroller support)

Wire	Description
<b>digital_out_initiate</b>	Initiate command (1 = Initiate, 0 = Abort)
<b>int_out_stop_count</b>	If < 1, no stop
<b>analog_in_1</b>	ADC channel 1 (volts)
<b>analog_in_2</b>	ADC channel 2 (volts)
<b>analog_in_3</b>	ADC channel 3 (volts)
<b>analog_in_4</b>	ADC channel 4 (volts)
<b>analog_in_5</b>	ADC channel 5 (volts)
<b>analog_in_6</b>	ADC channel 6 (volts)
<b>analog_in_7</b>	ADC channel 7 (volts)
<b>analog_in_8</b>	ADC channel 8 (volts)
<b>analog_out_1</b>	DAC channel 1 (volts)

<b>analog_out_2</b>	DAC channel 2 (volts)
<b>analog_out_3</b>	DAC channel 3 (volts)
<b>analog_out_4</b>	DAC channel 4 (volts)
<b>analog_out_5</b>	DAC channel 5 (volts)
<b>analog_out_6</b>	DAC channel 6 (volts)
<b>analog_out_7</b>	DAC channel 7 (volts)
<b>analog_out_8</b>	DAC channel 8 (volts)
<b>digital_in_1</b>	TTL digital input channel 1
<b>digital_in_2</b>	TTL digital input channel 2
<b>digital_in_3</b>	TTL digital input channel 3
<b>digital_in_4</b>	TTL digital input channel 4
<b>digital_in_5</b>	TTL digital input channel 5
<b>digital_in_6</b>	TTL digital input channel 6
<b>digital_in_7</b>	TTL digital input channel 7
<b>digital_in_8</b>	TTL digital input channel 8
<b>digital_out_1</b>	TTL digital output channel 1
<b>digital_out_2</b>	TTL digital output channel 2
<b>digital_out_3</b>	TTL digital output channel 3
<b>digital_out_4</b>	TTL digital output channel 4
<b>digital_out_5</b>	TTL digital output channel 5
<b>digital_out_6</b>	TTL digital output channel 6
<b>digital_out_7</b>	TTL digital output channel 7
<b>digital_out_8</b>	TTL digital output channel 8

### 5.18 N2400

<board type="N2400" ... > (Fiber slave device, G1 loop controller support only)

<b>Wire</b>	<b>Description</b>
<b>digital_in_mode_switch_n</b>	Mode switch position (n=1-24)
<b>digital_in_relay_switch_n</b>	Relay switch position (n=1-24)
<b>digital_in_limit_neg_n</b>	Limit switch neg state (n=1-24)
<b>digital_in_limit_pos_n</b>	Limit switch pos state (n=1-24)
<b>digital_out_switch_relay_n</b>	Switch relay (n=1-24)

## 6 Virtual Devices

( <interpreter> section )

Each device in this section is a custom or virtual device and is directly supported through IG2.

### 6.1 ASCIICLIENT

```
<asciiclient type="asciiclient" ip="TODO" port="TODO" proto="TODO" ... >
```

Wire	Description
<b>string_out_message</b>	String sent to device
<b>string_in_response</b>	Response received from device
<b>int_in_status</b>	TODO
<b>string_in_error_description</b>	TODO

### 6.2 BPM (Detector)

```
<detector type="bpm" ... >
```

**X AXIS = Channels 17-32**

**Y AXIS = Channels 1-16**

Wire	Description
<b>digital_out_position</b>	Position control (0=out, non-zero=in)
<b>analog_out_bias</b>	High voltage control (V)
<b>analog_in_bias</b>	High voltage readback (V)
<b>int_in_position</b>	Position readback (0=out, non-zero=in)
<b>analog_in_xcurrent_noscale</b>	X-axis current readback (A) This value = sum of the X axis channels.
<b>analog_in_ycurrent_noscale</b>	Y-axis current readback (A) This value = sum of the Y axis channels.
<b>analog_in_beamcurrent_noscale</b>	Beam current readback (A) This value = (sum of X axis channels + sum of Y axis channels) / 2
<b>analog_in_xcurrent</b>	X-axis current readback (variable units, autoscaled) See analog_in_xcurrent_noscale.
<b>analog_in_ycurrent</b>	Y-axis current readback (variable units, autoscaled) See analog_in_ycurrent_noscale.
<b>analog_in_beamcurrent</b>	Beam current readback (variable units, autoscaled) See analog_in_beamcurrent_noscale.
<b>variant_in_channels</b>	32-element array of channel data (A)
<b>analog_in_xpos_actual</b>	Calculated x-axis position readback (mm) Gaussabola algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_ypos_actual</b>	Calculated y-axis position readback (mm) Gaussabola algorithm. Returns -10000 when value cannot be calculated.

<b>analog_in_width_actual</b>	Calculated x-axis width readback (mm) Gaussabola algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_height_actual</b>	Calculated y-axes width readback (mm) Gaussabola algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_xpos_target</b>	Target x-axis position (not currently used)
<b>analog_in_ypos_target</b>	Target y-axis position (not currently used)
<b>analog_in_width_target</b>	Target x-axis width (not currently used)
<b>analog_in_height_target</b>	Target y-axis width (not currently used)

### 6.3 GAUSSIANFIT (Detector)

```
< detector type=" gaussianfit" name="TODO"
signal_threshold_percent ="#" max_signal_channels="#" weighting="#" num_peaks="#"
hcc="TODO" integrationtime="TODO" current="TODO" biasrdbk="TODO" biascmd="TODO"
initiate="TODO">
```

**num\_peaks** must be either 1 or 2.

If 2, the 128 channels are split in half and two gaussianfits are given.

The first fit is calculated from the first 64 channels, and the second fit is calculated from the last 64 channels.

Wire	Description
<b>digital_out_initiate</b>	Initiate I128
<b>analog_out_bias</b>	High voltage control (V)
<b>analog_in_bias</b>	High voltage readback (V)
<b>variant_in_channels</b>	128-element array of original channel data (A)
<b>analog_in_position</b>	Calculated position readback (mm). Center of Gravity algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_sigma</b>	Calculated sigma readback (mm). Center of Gravity algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_amplitude</b>	Calculated amplitude readback (mm). Gaussabola algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_confidence</b>	Confidence (%) -10000 = unable to get confidence
<b>analog[140]_in_gaussian_data</b>	[0...127] channel data (A) -20000 = unable to convert to A [128] HCC [129] Seconds (timestamp) [130] Nanoseconds (timestamp) [131...134] Over-range/under-range bits [135] Integration time [136] Position [137] Sigma [138] Amplitude [139] Confidence

<b>analog_in_position_2</b> <b>analog_in_sigma_2</b> <b>analog_in_amplitude_2</b> <b>analog_in_confidence_2</b> <b>analog[140]_in_gaussian_data_2</b>	Channels for second Gaussian fit Enabled if num_peaks==2
<b>int_out_confidence_window</b>	Confidence Signal Channels (Window)
<b>analog[5]_out_regression_options</b>	Configurable options for regression fit [0] = Signal Threshold Percent (Noise Threshold) [1] = Max Signal Channels (Window) [2] = Peak Threshold [3] = Sum Threshold [4] = Weighting

### 6.4 GCH30

<gch30 type="gch30" name="TODO" threshold\_pct="#" threshold\_fixed="#" flow\_cmd="TODO" flow\_rdbk="TODO" purge\_cmd="TODO" ... >

Wire	Description
<b>analog_out_flow</b>	Flow command
<b>analog_in_flow</b>	Flow readback
<b>digital_out_purge</b>	Purge command
<b>digital_in_alarm</b>	Alarm state

### 6.5 ILB ETH 24 DI16 DIO16-2TX

<ilbeth24 type="ilbeth24" name="TODO" ip="#" pollingPeriod="#" ... >

Wire	Description
<b>int_in_digital_io</b>	Read bits 0-15. They may be inputs or outputs, depending on the configuration
<b>int_in_digital_inputs</b>	Read bits 16-32
<b>int_out_digital_outputs</b>	Write bits 0-15 (if configured as outputs)
<b>int_in_status_register</b>	Bit 0 = 0: An error occurred Bit 0 = 1: No error Bit 1 = 0: No NetFail Bit 1 = 1: NetFail is present
<b>int_in_io_diagnostic_register</b>	Bit 0 = 1: Short circuit / output overload Bit 1 = 1: Short circuit / sensor supply overload Bit 2 = 1: Sensor supply U <sub>S1</sub> missing Bit 3 = 1: Sensor supply U <sub>S2</sub> missing
<b>int_in_netfail_reason</b>	See ILBETH24 datasheet for reason codes
<b>int_out_modbus_timeout</b>	Detects errors in the network or client 200-65000 (ms) 0 deactivates the monitoring system

<b>int_out_watchdog_timeout</b>	Timeout value for the process data watchdog 200-65000 (ms) 0 deactivates the watchdog
<b>int_out_fault_response_mode</b>	Setting or reading the fault response mode 0: All outputs are set to "0" 1: The digital outputs are set to "0" 2: All outputs retain their last value
<b>int_out_command_register</b>	0x02: NetFail acknowledgement 0x04: Diagnostic message acknowledgement (I/O error)

## 6.6 IONCHAMBER (Detector)

```
< detector type=" ionchamber" name="TODO" hcc="TODO" currentx="TODO" currenty="TODO"
biasrdbkx="TODO" biascmdx="TODO" biasrdbky="TODO" biascmdy="TODO" initiatex="TODO"
initiatey="TODO">
```

Wire	Description
<b>digital_out_initiate</b>	Initiate I128
<b>digital_out_position</b>	Position control (0=out, non-zero=in)
<b>analog_out_bias</b>	High voltage control (V)
<b>int_in_position</b>	Position readback (0=out, non-zero=in)
<b>analog_in_bias</b>	High voltage readback (V)
<b>analog_in_beamcurrent</b>	Beam current readback (A). This value = (sum of X axis channels + sum of Y axis channels) / 2
<b>analog_in_xcurrent</b>	X-axis current readback (A). This value = sum of the X axis channels.
<b>analog_in_ycurrent</b>	Y-axis current readback (A). This value = sum of the Y axis channels.
<b>variant_in_channelsx</b>	128-element array of channel data (A)
<b>variant_in_channelsy</b>	128-element array of channel data (A)
<b>analog_in_xpos_actual</b>	Calculated x-axis position readback (mm). Gaussabola algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_ypos_actual</b>	Calculated y-axis position readback (mm). Gaussabola algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_width_actual</b>	Calculated x-axis width readback (mm). Gaussabola algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_height_actual</b>	Calculated y-axes width readback (mm). Gaussabola algorithm. Returns -10000 when value cannot be calculated.
<b>analog_in_xpos_target</b>	Target x-axis position (not currently used)
<b>analog_in_ypos_target</b>	Target y-axis position (not currently used)
<b>analog_in_width_target</b>	Target x-axis width (not currently used)
<b>analog_in_height_target</b>	Target y-axis width (not currently used)

## 6.7 KEITHLEY

(manufacturing build only)

```
<keithley type="keithley" name="k238" id="0" address="16" >
```

Wire	Description
<b>int_out_sourcemode</b>	Source mode configuration (0=voltage, 1=current).
<b>analog_out_sourcelevel</b>	Source setpoint configuration.
<b>analog_out_compliancevoltage</b>	Compliance voltage configuration.
<b>digital_out_initiate</b>	Set operational mode (0=stop,1=run).
<b>digital_in_error</b>	Error state of last command sent (0=no error, 1=error).
<b>string_out_command</b>	Send string command. Max 40 characters.

## 6.8 KOLLMORGEN (MODBUS)

```
< kollmorgen type="kollmorgen" ip="TODO" pollingPeriod="#" ... >
```

Wire	Description
<b>analog_in_AIN_VALUE</b>	Read the value of the analog input signal
<b>int_out_DRV_CMDSOURCE</b>	Sets the command source
<b>int_out_DRV_OPMODE</b>	Sets the drive operation mode
<b>int_out_MT_CNTL</b>	Sets motion task control word; active in opmode 2 only
<b>int_out_MT_NUM</b>	Sets the motion task number; active in opmode 2 only
<b>int_out_MT_TNUM</b>	Sets the motion task customer table number; active in opmode 2 only
<b>digital_out_HOME_MOVE</b>	Starts a homing procedure; active in opmode 2 only
<b>digital_out_DRV_EN_DIS</b>	Enable/disable drive command 1: Enable 0: Disable
<b>digital_out_DRV_STOP</b>	Stop command 1: Execute command
<b>int_out_MOTOR_BRAKERLS</b>	Allows a user to release or apply the motor brake
<b>analog_out_MT_ACC</b>	Motion task acceleration
<b>analog_out_MT_DEC</b>	Motion task deceleration
<b>analog_out_MT_V</b>	Motion task velocity
<b>int_out_MT_MOVE</b>	Starts a motion task; active in opmode 2 only
<b>int_out_MT_SET</b>	Sets the motion task in the drive; active in opmode 2 only
<b>int_out_UNIT_PIN</b>	Sets gear IN for the unit conversion
<b>int_out_UNIT_POOUT</b>	Sets gear OUT for the unit conversion
<b>analog_out_MT_P</b>	Sets the motion task position; active in opmode 2 only
<b>analog_in_PL_FB</b>	Reads the position feedback value

<b>analog_in_VL_CMD</b>	Reads the actual velocity command; active in opmode 1 and opmode 2 only
<b>int_in_DRV_MOTIONSTAT</b>	Status bits: 0: Motion task is active 1: Home position found 2: Home routine finished 3: Homing active 4: Homing error occurred 5: Slave in electronic gearing 6: Electronic gearing active 7: Emergency stop in progress 8: Emergency stop procedure error 9: Service motion active 10: Motion task could not be activated 11: Motion task target position reached 12: Motion task target velocity reached 13: Motion task encountered an exception 14: Target position of motion task has been crossed 15: Actual position is within target position window 16: AKD Basic is executing a move 17: AKD Basic has completed a move 18: The fixed stop has been reached 19: The fixed stop has been passed 20: The axis broke off of the fixed position
<b>int_in_MODBUS_DIO</b>	Bits 0-7: Digital input ch.0 – ch.6 Bits 16-17: Digital output ch.0 – ch.1
<b>int_in_MODBUS_DRVSTAT</b>	Status Bits: 0: Drive active 1: STO status 2: Positive HW limit triggered 3: Negative HW limit triggered 4: Positive SW limit triggered 5: Negative SW limit triggered
<b>int_out_FB2_SOURCE</b>	Sets the source for the second feedback input 0 = None 1 = Feedback Source X9 2 = Feedback Source X7
<b>int_out_FB2_PIN</b>	Sets gear IN for FB2.P
<b>int_out_FB2_POUT</b>	Sets gear OUT for FB2.P
<b>analog_in_FB2_P</b>	Reads position from the secondary feedback
<b>analog_in_CAPO_PLFB</b>	Reads captured position value
<b>string_in_modbus_status</b>	Error from Modbus library Gets updated after each command
<b>string_in_kollmorgen_status</b>	Error from Kollmorgen Gets updated after each command

## 6.9 KOLLMORGEN (TELNET)

(Requires an AsciiClient)

```
< kollmorgentelnet type=" kollmorgentelnet" telnet_write_chn="TODO" telnet_read_chn="TODO"
telnet_status_chn="TODO" telnet_error_chn="TODO" ... >
```

Wire	Description
------	-------------

<b>analog_out_position</b>	MT.P (float)
<b>int_out_table</b>	MT.TNUM (integer)
<b>int_out_profile</b>	MT.CNTL (integer)
<b>int_out_task</b>	MT.NUM (integer)
<b>int_out_cmdsource</b>	DRV.CMDSOURCE (integer)
<b>int_out_opmode</b>	DRV.OPMODE (integer)
<b>digital_out_stop</b>	DRV.STOP (no argument)
<b>digital_out_move</b>	MT.MOV (no argument)
<b>digital_out_home</b>	HOME.MOV (no argument)
<b>digital_out_enable</b>	DRV.DIS + DRV.ENA (no argument)
<b>analog_in_position</b>	PL.FB
<b>int_in_status</b>	TODO
<b>string_in_error_description</b>	TODO

## 6.10 MEMBLOCK

<memblock type="memblock" size="n" ... >

Wire	Description
<b>analog_out_n</b>	General use analog output. (n=1-512)
<b>digital_out_n</b>	General use digital output. (n=1-512)
<b>int_out_n</b>	General use integer output. (n=1-512)
<b>string_out_n</b>	General use string output. (n=1-512)
<b>analog_in_n</b>	General use analog input. (n=1-512)
<b>digital_in_n</b>	General use digital input. (n=1-512)
<b>int_in_n</b>	General use integer input. (n=1-512)
<b>string_in_n</b>	General use string input. (n=1-512)

## 6.11 PIDCONTROL

<pidcontrol type="pidcontrol" name="TODO" pvChannel="TODO" outputChannel="TODO" period="100" updateRate="10" Kp="1000" Ki="0.01" Kd="0.0" outputMin="#" outputMax="#" enabled="1" ... >

**period:** How often PID is calculated (hz)

**updateRate:** How often outputChannel is updated (ms)

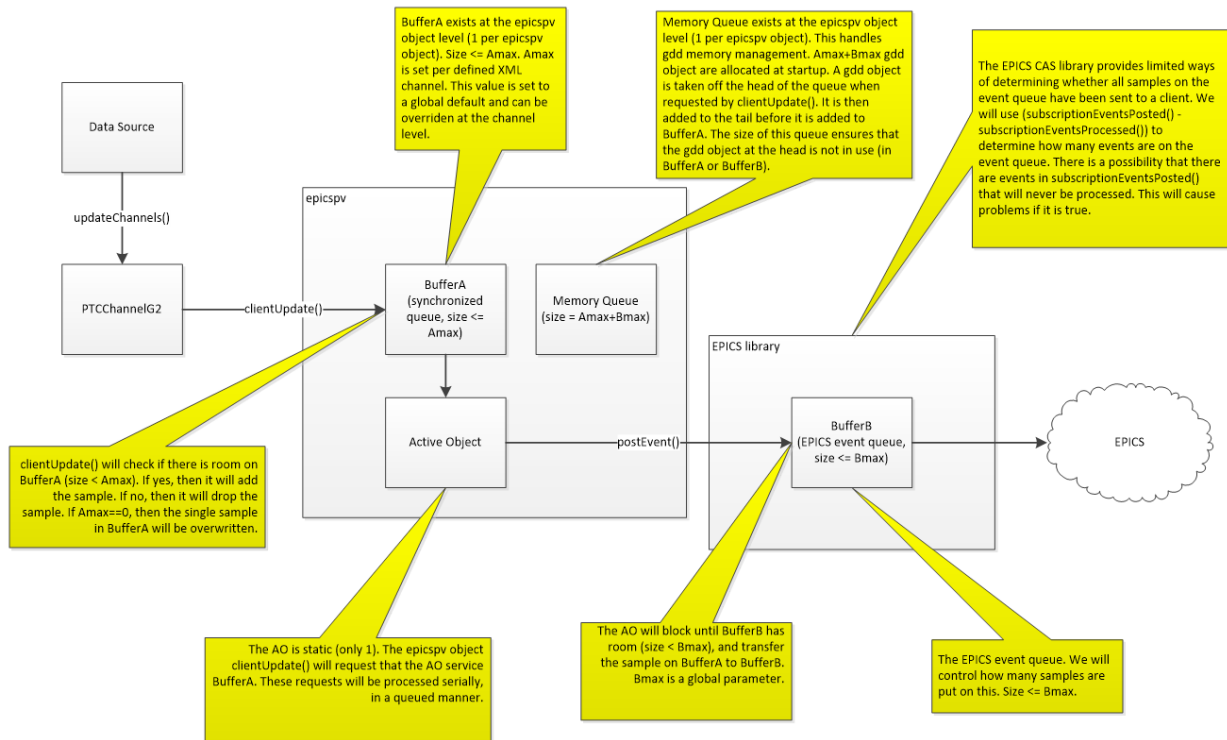
Wire	Description
<b>analog_out_setpoint</b>	Setpoint
<b>analog_out_kp</b>	Kp
<b>analog_out_ki</b>	Ki
<b>analog_out_kd</b>	Kd

<b>digital_out_enabled</b>	Enable or disable PID ( 1 = enabled, 0 = disabled)
----------------------------	---

# 7 Appendix 1 - Advanced Channel Buffering

## 7.1 Design

The EPICS event queue is prone to overflow and drop data due to high-rate updates from IG2 objects. This diagram illustrates the buffering technique used to prevent the event queue from overflowing, and prevent data loss.  
C. Pendleton Sept. 19, 2014



## 7.2 Global Options

There are 3 global buffering options:

```
<epicscas name="ecas" aMax="10" bMax="20" monitorOnlyChange="true"/>
```

- **aMax** is an optional parameter that determines the global value that all channels will default to. This can be overridden at the channel level (see 7.1.2). The default value of this parameter is 0 (unbuffered).
- **bMax** determines the maximum allowed number of samples on the EPICS event queue. Changing this is not recommended. The default value is 10.
- **monitorOnlyChange** is an optional parameter that determines the global value that all channels will default to. It can be overridden at the channel level (see 7.1.4). The default value of this parameter is false.

## 7.3 Reading Timeslice Database from RTP file

If you are pushing down an xml file to do calculations in the realtime processor, there may be a `<timeslicedatabase>` section. This data can be read using the `analog[256]_in_timeslice_data` channel (if available on that device). The `<timeslicedatabase>` structure is converted to all analog values and sent to this channel. Make sure you convert any analog values back to the type you expect. If the `<timeslicedatabase>` structure is less than 256 values, the rest will be filled in with zeros.