Previous topic
gnuradio.wxgui
Next topic
gnuradio.analog
This Page
Show Source
Quick search
Go
Enter search terms or a module, class or function name.

gnuradio.zeromq

Blocks for interfacing with ZeroMQ endpoints.

gnuradio.zeromq.**pub_msg_sink**(*char * address*, *int timeout=100*) → pub_msg_sink_sptr

Sink the contents of a msg port to a ZMQ PUB socket.

This block acts a message port receiver and writes individual messages to a ZMQ PUB socket. A PUB socket may have subscribers and will pass all incoming messages to each subscriber. Subscribers can be either another gr-zeromq source block or a non-GNU Radio ZMQ socket.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of zeromq::pub_msg_sink.

Parameters:
- **address** – ZMQ socket address specifier
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments

pub_msg_sink_sptr.**active_thread_priority**(*pub_msg_sink_sptr self*) → int

pub_msg_sink_sptr.**declare_sample_delay**(*pub_msg_sink_sptr self*, *int which*, *int delay*)

declare_sample_delay(pub_msg_sink_sptr self, unsigned int delay)

pub_msg_sink_sptr.**message_subscribers**(*pub_msg_sink_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

pub_msg_sink_sptr.**min_noutput_items**(*pub_msg_sink_sptr self*) → int

pub_msg_sink_sptr.**pc_input_buffers_full_avg**(*pub_msg_sink_sptr self*, *int which*) → float

pc_input_buffers_full_avg(pub_msg_sink_sptr self) -> pmt_vector_float

pub_msg_sink_sptr.**pc_noutput_items_avg**(*pub_msg_sink_sptr self*) → float

pub_msg_sink_sptr.**pc_nproduced_avg**(*pub_msg_sink_sptr self*) → float

pub_msg_sink_sptr.**pc_output_buffers_full_avg**(*pub_msg_sink_sptr self*, *int which*) → float

pc_output_buffers_full_avg(pub_msg_sink_sptr self) -> pmt_vector_float

pub_msg_sink_sptr.**pc_throughput_avg**(*pub_msg_sink_sptr self*) → float

pub_msg_sink_sptr.**pc_work_time_avg**(*pub_msg_sink_sptr self*) → float

pub_msg_sink_sptr.**pc_work_time_total**(*pub_msg_sink_sptr self*) → float

pub_msg_sink_sptr.**sample_delay**(*pub_msg_sink_sptr self*, *int which*) → unsigned int

pub_msg_sink_sptr.**set_min_noutput_items**(*pub_msg_sink_sptr self*, *int m*)

pub_msg_sink_sptr.**set_thread_priority**(*pub_msg_sink_sptr self*, *int priority*) → int

pub_msg_sink_sptr.**thread_priority**(*pub_msg_sink_sptr self*) → int

gnuradio.zeromq.**pub_sink**(*size_t itemsize*, *size_t vlen*, *char * address*, *int timeout=100*, *bool pass_tags=False*, *int hwm=-1*) → pub_sink_sptr

Sink the contents of a stream to a ZMQ PUB socket.

This block acts a a streaming sink for a GNU Radio flowgraph and writes its contents to a ZMQ PUB socket. A PUB socket may have subscribers and will pass all incoming stream data to each subscriber. Subscribers can be either another gr-zeromq source block or a non-GNU Radio ZMQ socket.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of zeromq::pub_sink.

Parameters:

- **itemsize** – Size of a stream item in bytes.
- **vlen** – Vector length of the input items. Note that one vector is one item.
- **address** – ZMQ socket address specifier.
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments.
- **pass_tags** – Whether sink will serialize and pass tags over the link.
- **hwm** – High Watermark to configure the socket to (-1 => zmq's default)

pub_sink_sptr.**active_thread_priority**(*pub_sink_sptr self*) → int

pub_sink_sptr.**declare_sample_delay**(*pub_sink_sptr self*, *int which*, *int delay*)

declare_sample_delay(pub_sink_sptr self, unsigned int delay)

pub_sink_sptr.**message_subscribers**(*pub_sink_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

pub_sink_sptr.**min_noutput_items**(*pub_sink_sptr self*) → int

pub_sink_sptr.**pc_input_buffers_full_avg**(*pub_sink_sptr self*, *int which*) → float

pc_input_buffers_full_avg(pub_sink_sptr self) -> pmt_vector_float

pub_sink_sptr.**pc_noutput_items_avg**(*pub_sink_sptr self*) → float

pub_sink_sptr.**pc_nproduced_avg**(*pub_sink_sptr self*) → float

pub_sink_sptr.**pc_output_buffers_full_avg**(*pub_sink_sptr self*, *int which*) → float

pc_output_buffers_full_avg(pub_sink_sptr self) -> pmt_vector_float

pub_sink_sptr.**pc_throughput_avg**(*pub_sink_sptr self*) → float

pub_sink_sptr.**pc_work_time_avg**(*pub_sink_sptr self*) → float

pub_sink_sptr.**pc_work_time_total**(*pub_sink_sptr self*) → float

pub_sink_sptr.**sample_delay**(*pub_sink_sptr self*, *int which*) → unsigned int

pub_sink_sptr.**set_min_noutput_items**(*pub_sink_sptr self*, *int m*)

pub_sink_sptr.**set_thread_priority**(*pub_sink_sptr self*, *int priority*) → int

pub_sink_sptr.**thread_priority**(*pub_sink_sptr self*) → int

gnuradio.zeromq.**pull_msg_source**(*char * address*, *int timeout=100*) → pull_msg_source_sptr

Receive messages on ZMQ PULL socket and output async messages.

This block will connect to a ZMQ PUSH socket, then convert received messages to outgoing async messages.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of gr::zeromq::pull_msg_source.

Parameters:
- **address** – ZMQ socket address specifier
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments

pull_msg_source_sptr.**active_thread_priority**(*pull_msg_source_sptr self*) → int

pull_msg_source_sptr.**declare_sample_delay**(*pull_msg_source_sptr self*, *int which*, *int delay*)

declare_sample_delay(pull_msg_source_sptr self, unsigned int delay)

pull_msg_source_sptr.**message_subscribers**(*pull_msg_source_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

pull_msg_source_sptr.**min_noutput_items**(*pull_msg_source_sptr self*) → int

pull_msg_source_sptr.**pc_input_buffers_full_avg**(*pull_msg_source_sptr self*, *int which*) → float

pc_input_buffers_full_avg(pull_msg_source_sptr self) -> pmt_vector_float

pull_msg_source_sptr.**pc_noutput_items_avg**(*pull_msg_source_sptr self*) → float

pull_msg_source_sptr.**pc_nproduced_avg**(*pull_msg_source_sptr self*) → float

pull_msg_source_sptr.**pc_output_buffers_full_avg**(*pull_msg_source_sptr self*, *int which*) → float

pc_output_buffers_full_avg(pull_msg_source_sptr self) -> pmt_vector_float

pull_msg_source_sptr.**pc_throughput_avg**(*pull_msg_source_sptr self*) → float

pull_msg_source_sptr.**pc_work_time_avg**(*pull_msg_source_sptr self*) → float

pull_msg_source_sptr.**pc_work_time_total**(*pull_msg_source_sptr self*) → float

pull_msg_source_sptr.**sample_delay**(*pull_msg_source_sptr self*, *int which*) → unsigned int

pull_msg_source_sptr.**set_min_noutput_items**(*pull_msg_source_sptr self*, *int m*)

pull_msg_source_sptr.**set_thread_priority**(*pull_msg_source_sptr self*, *int priority*) → int

pull_msg_source_sptr.**thread_priority**(*pull_msg_source_sptr self*) → int

gnuradio.zeromq.**pull_source**(*size_t itemsize*, *size_t vlen*, *char * address*, *int timeout=100*, *bool pass_tags=False*, *int hwm=-1*) → pull_source_sptr

Receive messages on ZMQ PULL socket and source stream.

This block will connect to a ZMQ PUSH socket, then produce all incoming messages as streaming output.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of gr::zeromq::pull_source.

Parameters:
- **itemsize** – Size of a stream item in bytes.
- **vlen** – Vector length of the input items. Note that one vector is one item.
- **address** – ZMQ socket address specifier.
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments.
- **pass_tags** – Whether source will look for and deserialize tags.
- **hwm** – High Watermark to configure the socket to (-1 => zmq's default)

pull_source_sptr.**active_thread_priority**(*pull_source_sptr self*) → int

pull_source_sptr.**declare_sample_delay**(*pull_source_sptr self*, *int which*, *int delay*)

declare_sample_delay(pull_source_sptr self, unsigned int delay)

pull_source_sptr.**message_subscribers**(*pull_source_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

pull_source_sptr.**min_noutput_items**(*pull_source_sptr self*) → int

pull_source_sptr.**pc_input_buffers_full_avg**(*pull_source_sptr self*, *int which*) → float

pc_input_buffers_full_avg(pull_source_sptr self) -> pmt_vector_float

pull_source_sptr.**pc_noutput_items_avg**(*pull_source_sptr self*) → float

pull_source_sptr.**pc_nproduced_avg**(*pull_source_sptr self*) → float

pull_source_sptr.**pc_output_buffers_full_avg**(*pull_source_sptr self*, *int which*) → float

pc_output_buffers_full_avg(pull_source_sptr self) -> pmt_vector_float

pull_source_sptr.**pc_throughput_avg**(*pull_source_sptr self*) → float

pull_source_sptr.**pc_work_time_avg**(*pull_source_sptr self*) → float

pull_source_sptr.**pc_work_time_total**(*pull_source_sptr self*) → float

pull_source_sptr.**sample_delay**(*pull_source_sptr self*, *int which*) → unsigned int

pull_source_sptr.**set_min_noutput_items**(*pull_source_sptr self*, *int m*)

pull_source_sptr.**set_thread_priority**(*pull_source_sptr self*, *int priority*) → int

pull_source_sptr.**thread_priority**(*pull_source_sptr self*) → int

gnuradio.zeromq.**push_msg_sink**(*char * address*, *int timeout=100*) → push_msg_sink_sptr

Sink the contents of a msg port to a ZMQ PUSH socket.

This block acts a message port receiver and writes individual messages to a ZMQ PUSH socket. The corresponding receiving ZMQ PULL socket can be either another gr-zeromq source block or a non-GNU Radio ZMQ socket.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of gr::zeromq::push_msg_sink.

Parameters:
- **address** – ZMQ socket address specifier
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments

push_msg_sink_sptr.**active_thread_priority**(*push_msg_sink_sptr self*) →

int

push_msg_sink_sptr.**declare_sample_delay**(*push_msg_sink_sptr self*, *int which*, *int delay*)

declare_sample_delay(push_msg_sink_sptr self, unsigned int delay)

push_msg_sink_sptr.**message_subscribers**(*push_msg_sink_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

push_msg_sink_sptr.**min_noutput_items**(*push_msg_sink_sptr self*) → int

push_msg_sink_sptr.**pc_input_buffers_full_avg**(*push_msg_sink_sptr self*, *int which*) → float

pc_input_buffers_full_avg(push_msg_sink_sptr self) -> pmt_vector_float

push_msg_sink_sptr.**pc_noutput_items_avg**(*push_msg_sink_sptr self*) → float

push_msg_sink_sptr.**pc_nproduced_avg**(*push_msg_sink_sptr self*) → float

push_msg_sink_sptr.**pc_output_buffers_full_avg**(*push_msg_sink_sptr self*, *int which*) → float

pc_output_buffers_full_avg(push_msg_sink_sptr self) -> pmt_vector_float

push_msg_sink_sptr.**pc_throughput_avg**(*push_msg_sink_sptr self*) → float

push_msg_sink_sptr.**pc_work_time_avg**(*push_msg_sink_sptr self*) → float

push_msg_sink_sptr.**pc_work_time_total**(*push_msg_sink_sptr self*) → float

push_msg_sink_sptr.**sample_delay**(*push_msg_sink_sptr self*, *int which*) → unsigned int

push_msg_sink_sptr.**set_min_noutput_items**(*push_msg_sink_sptr self*, *int m*)

push_msg_sink_sptr.**set_thread_priority**(*push_msg_sink_sptr self*, *int priority*) → int

push_msg_sink_sptr.**thread_priority**(*push_msg_sink_sptr self*) → int

gnuradio.zeromq.**push_sink**(*size_t itemsize*, *size_t vlen*, *char * address*, *int timeout=100*, *bool pass_tags=False*, *int hwm=-1*) → push_sink_sptr

Sink the contents of a stream to a ZMQ PUSH socket.

This block acts a a streaming sink for a GNU Radio flowgraph and writes its contents to a ZMQ PUSH socket. A PUSH socket will round-robin send its messages to each connected ZMQ PULL socket, either another gr-zeromq source block or a regular, non-GNU Radio ZMQ socket.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of gr::zeromq::push_sink.

Parameters:

- **itemsize** – Size of a stream item in bytes.
- **vlen** – Vector length of the input items. Note that one vector is one item.
- **address** – ZMQ socket address specifier.
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments.
- **pass_tags** – Whether sink will serialize and pass tags over the link.
- **hwm** – High Watermark to configure the socket to (-1 => zmq's default)

push_sink_sptr.**active_thread_priority**(*push_sink_sptr self*) → int

push_sink_sptr.**declare_sample_delay**(*push_sink_sptr self*, *int which*, *int delay*)

declare_sample_delay(push_sink_sptr self, unsigned int delay)

push_sink_sptr.**message_subscribers**(*push_sink_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

push_sink_sptr.**min_noutput_items**(*push_sink_sptr self*) → int

push_sink_sptr.**pc_input_buffers_full_avg**(*push_sink_sptr self*, *int which*) → float

pc_input_buffers_full_avg(push_sink_sptr self) -> pmt_vector_float

push_sink_sptr.**pc_noutput_items_avg**(*push_sink_sptr self*) → float

push_sink_sptr.**pc_nproduced_avg**(*push_sink_sptr self*) → float

push_sink_sptr.**pc_output_buffers_full_avg**(*push_sink_sptr self*, *int which*) → float

pc_output_buffers_full_avg(push_sink_sptr self) -> pmt_vector_float

push_sink_sptr.**pc_throughput_avg**(*push_sink_sptr self*) → float

push_sink_sptr.**pc_work_time_avg**(*push_sink_sptr self*) → float

push_sink_sptr.**pc_work_time_total**(*push_sink_sptr self*) → float

push_sink_sptr.**sample_delay**(*push_sink_sptr self*, *int which*) → unsigned int

push_sink_sptr.**set_min_noutput_items**(*push_sink_sptr self*, *int m*)

push_sink_sptr.**set_thread_priority**(*push_sink_sptr self*, *int priority*) → int

push_sink_sptr.**thread_priority**(*push_sink_sptr self*) → int

gnuradio.zeromq.**rep_msg_sink**(*char * address*, *int timeout=100*) → rep_msg_sink_sptr

Sink the contents of a msg port to a ZMQ REP socket.

This block acts a message port receiver and writes individual messages to a ZMQ REP socket. The corresponding receiving ZMQ REQ socket can be either another gr-zeromq source block or a non-GNU Radio ZMQ socket.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of zeromq::rep_msg_sink.

Parameters:
- **address** – ZMQ socket address specifier
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments

rep_msg_sink_sptr.**active_thread_priority**(*rep_msg_sink_sptr self*) → int

rep_msg_sink_sptr.**declare_sample_delay**(*rep_msg_sink_sptr self*, *int which*, *int delay*)

declare_sample_delay(rep_msg_sink_sptr self, unsigned int delay)

rep_msg_sink_sptr.**message_subscribers**(*rep_msg_sink_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

rep_msg_sink_sptr.**min_noutput_items**(*rep_msg_sink_sptr self*) → int

rep_msg_sink_sptr.**pc_input_buffers_full_avg**(*rep_msg_sink_sptr self*, *int which*) → float

pc_input_buffers_full_avg(rep_msg_sink_sptr self) -> pmt_vector_float

rep_msg_sink_sptr.**pc_noutput_items_avg**(*rep_msg_sink_sptr self*) → float

rep_msg_sink_sptr.**pc_nproduced_avg**(*rep_msg_sink_sptr self*) → float

rep_msg_sink_sptr.**pc_output_buffers_full_avg**(*rep_msg_sink_sptr self*, *int which*) → float

pc_output_buffers_full_avg(rep_msg_sink_sptr self) -> pmt_vector_float

rep_msg_sink_sptr.**pc_throughput_avg**(*rep_msg_sink_sptr self*) → float

rep_msg_sink_sptr.**pc_work_time_avg**(*rep_msg_sink_sptr self*) → float

rep_msg_sink_sptr.**pc_work_time_total**(*rep_msg_sink_sptr self*) → float

rep_msg_sink_sptr.**sample_delay**(*rep_msg_sink_sptr self*, *int which*) → unsigned int

rep_msg_sink_sptr.**set_min_noutput_items**(*rep_msg_sink_sptr self*, *int m*)

rep_msg_sink_sptr.**set_thread_priority**(*rep_msg_sink_sptr self*, *int priority*) → int

rep_msg_sink_sptr.**thread_priority**(*rep_msg_sink_sptr self*) → int

gnuradio.zeromq.**rep_sink**(*size_t itemsize*, *size_t vlen*, *char * address*, *int timeout=100*, *bool pass_tags=False*, *int hwm=-1*) → rep_sink_sptr

Sink the contents of a stream to a ZMQ REP socket.

This block acts a a streaming sink for a GNU Radio flowgraph and writes its contents to a ZMQ REP socket. A REP socket will only send its contents to an attached REQ socket when it requests items.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of zeromq::rep_sink.

Parameters:

- **itemsize** – Size of a stream item in bytes.
- **vlen** – Vector length of the input items. Note that one vector is one item.
- **address** – ZMQ socket address specifier.
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments.
- **pass_tags** – Whether sink will serialize and pass tags over the link.
- **hwm** – High Watermark to configure the socket to (-1 => zmq's default)

rep_sink_sptr.**active_thread_priority**(*rep_sink_sptr self*) → int

rep_sink_sptr.**declare_sample_delay**(*rep_sink_sptr self*, *int which*, *int delay*)

declare_sample_delay(rep_sink_sptr self, unsigned int delay)

rep_sink_sptr.**message_subscribers**(*rep_sink_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

rep_sink_sptr.**min_noutput_items**(*rep_sink_sptr self*) → int

rep_sink_sptr.**pc_input_buffers_full_avg**(*rep_sink_sptr self*, *int which*) → float

pc_input_buffers_full_avg(rep_sink_sptr self) -> pmt_vector_float

rep_sink_sptr.**pc_noutput_items_avg**(*rep_sink_sptr self*) → float

rep_sink_sptr.**pc_nproduced_avg**(*rep_sink_sptr self*) → float

rep_sink_sptr.**pc_output_buffers_full_avg**(*rep_sink_sptr self*, *int which*) → float

pc_output_buffers_full_avg(rep_sink_sptr self) -> pmt_vector_float

rep_sink_sptr.**pc_throughput_avg**(*rep_sink_sptr self*) → float

rep_sink_sptr.**pc_work_time_avg**(*rep_sink_sptr self*) → float

rep_sink_sptr.**pc_work_time_total**(*rep_sink_sptr self*) → float

rep_sink_sptr.**sample_delay**(*rep_sink_sptr self*, *int which*) → unsigned int

rep_sink_sptr.**set_min_noutput_items**(*rep_sink_sptr self*, *int m*)

rep_sink_sptr.**set_thread_priority**(*rep_sink_sptr self*, *int priority*) → int

rep_sink_sptr.**thread_priority**(*rep_sink_sptr self*) → int

gnuradio.zeromq.**req_msg_source**(*char * address*, *int timeout=100*) → req_msg_source_sptr

Receive messages on ZMQ REQ socket output async messages.

This block will connect to a ZMQ REP socket, then resend all incoming messages as asynchronous messages.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of zeromq::req_msg_source.

Parameters:
- **address** – ZMQ socket address specifier
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments

req_msg_source_sptr.**active_thread_priority**(*req_msg_source_sptr self*) → int

req_msg_source_sptr.**declare_sample_delay**(*req_msg_source_sptr self*, *int which*, *int delay*)

declare_sample_delay(req_msg_source_sptr self, unsigned int delay)

req_msg_source_sptr.**message_subscribers**(*req_msg_source_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

req_msg_source_sptr.**min_noutput_items**(*req_msg_source_sptr self*) → int

req_msg_source_sptr.**pc_input_buffers_full_avg**(*req_msg_source_sptr self*, *int which*) → float

pc_input_buffers_full_avg(req_msg_source_sptr self) -> pmt_vector_float

req_msg_source_sptr.**pc_noutput_items_avg**(*req_msg_source_sptr self*) → float

req_msg_source_sptr.**pc_nproduced_avg**(*req_msg_source_sptr self*) → float

req_msg_source_sptr.**pc_output_buffers_full_avg**(*req_msg_source_sptr self*, *int which*) → float

pc_output_buffers_full_avg(req_msg_source_sptr self) -> pmt_vector_float

req_msg_source_sptr.**pc_throughput_avg**(*req_msg_source_sptr self*) → float

req_msg_source_sptr.**pc_work_time_avg**(*req_msg_source_sptr self*) → float

req_msg_source_sptr.**pc_work_time_total**(*req_msg_source_sptr self*) → float

req_msg_source_sptr.**sample_delay**(*req_msg_source_sptr self*, *int which*) →

unsigned int

req_msg_source_sptr.**set_min_noutput_items**(*req_msg_source_sptr self*, *int m*)

req_msg_source_sptr.**set_thread_priority**(*req_msg_source_sptr self*, *int priority*) → int

req_msg_source_sptr.**thread_priority**(*req_msg_source_sptr self*) → int

gnuradio.zeromq.**req_source**(*size_t itemsize*, *size_t vlen*, *char * address*, *int timeout=100*, *bool pass_tags=False*, *int hwm=-1*) → req_source_sptr

Receive messages on ZMQ REQ socket and source stream.

This block will connect to a ZMQ REP socket, then produce all incoming messages as streaming output.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of zeromq::req_source.

Parameters:
- **itemsize** – Size of a stream item in bytes.
- **vlen** – Vector length of the input items. Note that one vector is one item.
- **address** – ZMQ socket address specifier.
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments.
- **pass_tags** – Whether source will look for and deserialize tags.
- **hwm** – High Watermark to configure the socket to (-1 => zmq's default)

req_source_sptr.**active_thread_priority**(*req_source_sptr self*) → int

req_source_sptr.**declare_sample_delay**(*req_source_sptr self*, *int which*, *int delay*)

declare_sample_delay(req_source_sptr self, unsigned int delay)

req_source_sptr.**message_subscribers**(*req_source_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

req_source_sptr.**min_noutput_items**(*req_source_sptr self*) → int

req_source_sptr.**pc_input_buffers_full_avg**(*req_source_sptr self*, *int which*) → float

pc_input_buffers_full_avg(req_source_sptr self) -> pmt_vector_float

req_source_sptr.**pc_noutput_items_avg**(*req_source_sptr self*) → float

req_source_sptr.**pc_nproduced_avg**(*req_source_sptr self*) → float

req_source_sptr.**pc_output_buffers_full_avg**(*req_source_sptr self*, *int which*) → float

pc_output_buffers_full_avg(req_source_sptr self) -> pmt_vector_float

req_source_sptr.**pc_throughput_avg**(*req_source_sptr self*) → float

req_source_sptr.**pc_work_time_avg**(*req_source_sptr self*) → float

req_source_sptr.**pc_work_time_total**(*req_source_sptr self*) → float

req_source_sptr.**sample_delay**(*req_source_sptr self*, *int which*) → unsigned int

req_source_sptr.**set_min_noutput_items**(*req_source_sptr self*, *int m*)

req_source_sptr.**set_thread_priority**(*req_source_sptr self*, *int priority*) → int

req_source_sptr.**thread_priority**(*req_source_sptr self*) → int

gnuradio.zeromq.**sub_msg_source**(*char * address*, *int timeout=100*) → sub_msg_source_sptr

Receive messages on ZMQ SUB socket and output async messages.

This block will connect to a ZMQ PUB socket, then convert them to outgoing async messages

Constructor Specific Documentation:

Return a shared_ptr to a new instance of gr::zeromq::sub_msg_source.

Parameters:
- **address** – ZMQ socket address specifier
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments

sub_msg_source_sptr.**active_thread_priority**(*sub_msg_source_sptr self*) → int

sub_msg_source_sptr.**declare_sample_delay**(*sub_msg_source_sptr self*, *int which*, *int delay*)

declare_sample_delay(sub_msg_source_sptr self, unsigned int delay)

sub_msg_source_sptr.**message_subscribers**(*sub_msg_source_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

sub_msg_source_sptr.**min_noutput_items**(*sub_msg_source_sptr self*) → int

sub_msg_source_sptr.**pc_input_buffers_full_avg**(*sub_msg_source_sptr self*, *int which*) → float

pc_input_buffers_full_avg(sub_msg_source_sptr self) -> pmt_vector_float

sub_msg_source_sptr.**pc_noutput_items_avg**(*sub_msg_source_sptr self*) → float

sub_msg_source_sptr.**pc_nproduced_avg**(*sub_msg_source_sptr self*) → float

sub_msg_source_sptr.**pc_output_buffers_full_avg**(*sub_msg_source_sptr self*, *int which*) → float

pc_output_buffers_full_avg(sub_msg_source_sptr self) -> pmt_vector_float

sub_msg_source_sptr.**pc_throughput_avg**(*sub_msg_source_sptr self*) → float

sub_msg_source_sptr.**pc_work_time_avg**(*sub_msg_source_sptr self*) → float

sub_msg_source_sptr.**pc_work_time_total**(*sub_msg_source_sptr self*) → float

sub_msg_source_sptr.**sample_delay**(*sub_msg_source_sptr self*, *int which*) → unsigned int

sub_msg_source_sptr.**set_min_noutput_items**(*sub_msg_source_sptr self*, *int m*)

sub_msg_source_sptr.**set_thread_priority**(*sub_msg_source_sptr self*, *int priority*) → int

sub_msg_source_sptr.**thread_priority**(*sub_msg_source_sptr self*) → int

gnuradio.zeromq.**sub_source**(*size_t itemsize*, *size_t vlen*, *char * address*, *int timeout=100*, *bool pass_tags=False*, *int hwm=-1*) → sub_source_sptr

Receive messages on ZMQ SUB socket and source stream.

This block will connect to a ZMQ PUB socket, then produce all incoming messages as streaming output.

Constructor Specific Documentation:

Return a shared_ptr to a new instance of gr::zeromq::sub_source.

Parameters:

- **itemsize** – Size of a stream item in bytes.
- **vlen** – Vector length of the input items. Note that one vector is one item.
- **address** – ZMQ socket address specifier.
- **timeout** – Receive timeout in milliseconds, default is 100ms, 1us increments.
- **pass_tags** – Whether source will look for and deserialize tags.
- **hwm** – High Watermark to configure the socket to (-1 => zmq's default)

sub_source_sptr.**active_thread_priority**(*sub_source_sptr self*) → int

sub_source_sptr.**declare_sample_delay**(*sub_source_sptr self*, *int which*, *int delay*)

declare_sample_delay(sub_source_sptr self, unsigned int delay)

sub_source_sptr.**message_subscribers**(*sub_source_sptr self*, *swig_int_ptr which_port*) → swig_int_ptr

sub_source_sptr.**min_noutput_items**(*sub_source_sptr self*) → int

sub_source_sptr.**pc_input_buffers_full_avg**(*sub_source_sptr self*, *int which*) → float

pc_input_buffers_full_avg(sub_source_sptr self) -> pmt_vector_float

sub_source_sptr.**pc_noutput_items_avg**(*sub_source_sptr self*) → float

sub_source_sptr.**pc_nproduced_avg**(*sub_source_sptr self*) → float

sub_source_sptr.**pc_output_buffers_full_avg**(*sub_source_sptr self*, *int which*) → float

pc_output_buffers_full_avg(sub_source_sptr self) -> pmt_vector_float

sub_source_sptr.**pc_throughput_avg**(*sub_source_sptr self*) → float

sub_source_sptr.**pc_work_time_avg**(*sub_source_sptr self*) → float

sub_source_sptr.**pc_work_time_total**(*sub_source_sptr self*) → float

sub_source_sptr.**sample_delay**(*sub_source_sptr self*, *int which*) → unsigned int

sub_source_sptr.**set_min_noutput_items**(*sub_source_sptr self*, *int m*)

sub_source_sptr.**set_thread_priority**(*sub_source_sptr self*, *int priority*) → int

sub_source_sptr.**thread_priority**(*sub_source_sptr self*) → int