Previous topic

gnuradio.channels

Next topic

gnuradio.digital

This Page

Show Source

Quick search

Go

Enter search terms or a module, class or function name.

gnuradio.comedi

Blocks and utilities for COMEDI devices

gnuradio.comedi.**sink_s**(*int sampling_freq*, *std::string const dev*) → sink_s_sptr

sink using COMEDI

The sink has one input stream of signed short integers.

Input samples must be in the range [-32768,32767].

Constructor Specific Documentation:

make a COMEDI sink.

Parameters:
- **sampling_freq** – sampling rate in Hz
- **dev** – COMEDI device name, e.g., “/dev/comedi0”

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

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

declare_sample_delay(sink_s_sptr self, unsigned int delay)

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

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

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

pc_input_buffers_full_avg(sink_s_sptr self) -> pmt_vector_float

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

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

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

pc_output_buffers_full_avg(sink_s_sptr self) -> pmt_vector_float

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

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

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

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

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

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

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

gnuradio.comedi.**source_s**(*int sampling_freq*, *std::string const dev*) → source_s_sptr

source using COMEDI

The source has one to many input stream of signed short integers.

Output samples will be in the range [-32768,32767].

Constructor Specific Documentation:

make a COMEDI source.

Parameters:

- **sampling_freq** – sampling rate in Hz
- **dev** – COMEDI device name, e.g., “/dev/comedi0”

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

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

declare_sample_delay(source_s_sptr self, unsigned int delay)

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

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

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

pc_input_buffers_full_avg(source_s_sptr self) -> pmt_vector_float

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

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

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

pc_output_buffers_full_avg(source_s_sptr self) -> pmt_vector_float

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

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

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

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

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

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

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