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

gnuradio.fft

Fourier-transform blocks and related functions.

gnuradio.fft.**ctrlport_probe_psd**(*std::string const & id*, *std::string const & desc*, *int len*) → ctrlport_probe_psd_sptr

A ControlPort probe to export vectors of signals.

This block acts as a sink in the flowgraph but also exports vectors of complex samples over ControlPort. This block holds the latestnumber of complex samples so that every query by a ControlPort client will get the same length vector.

Constructor Specific Documentation:

Make a ControlPort probe block.

Parameters:
- **id** – A string ID to name the probe over ControlPort.
- **desc** – A string describing the probe.
- **len** – Number of samples to transmit.

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

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

declare_sample_delay(ctrlport_probe_psd_sptr self, unsigned int delay)

ctrlport_probe_psd_sptr.**get**(*ctrlport_probe_psd_sptr self*) → pmt_vector_cfloat

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

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

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

pc_input_buffers_full_avg(ctrlport_probe_psd_sptr self) -> pmt_vector_float

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

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

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

pc_output_buffers_full_avg(ctrlport_probe_psd_sptr self) -> pmt_vector_float

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

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

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

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

ctrlport_probe_psd_sptr.**set_length**(*ctrlport_probe_psd_sptr self*, *int len*)

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

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

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

gnuradio.fft.**fft_vcc**(*int fft_size*, *bool forward*, *pmt_vector_float window*, *bool shift=False*, *int nthreads=1*) → fft_vcc_sptr

Compute forward or reverse FFT. complex vector in / complex vector out.

Constructor Specific Documentation:

Parameters:
- **fft_size** –
- **forward** –
- **window** –
- **shift** –
- **nthreads** –

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

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

declare_sample_delay(fft_vcc_sptr self, unsigned int delay)

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

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

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

pc_input_buffers_full_avg(fft_vcc_sptr self) -> pmt_vector_float

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

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

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

pc_output_buffers_full_avg(fft_vcc_sptr self) -> pmt_vector_float

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

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

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

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

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

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

fft_vcc_sptr.**set_window**(*fft_vcc_sptr self*, *pmt_vector_float window*) → bool

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

gnuradio.fft.**fft_vfc**(*int fft_size*, *bool forward*, *pmt_vector_float window*, *int nthreads=1*) → fft_vfc_sptr

Compute forward or reverse FFT. float vector in / complex vector out.

Constructor Specific Documentation:

Parameters:
- **fft_size** –
- **forward** –
- **window** –
- **nthreads** –

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

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

declare_sample_delay(fft_vfc_sptr self, unsigned int delay)

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

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

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

pc_input_buffers_full_avg(fft_vfc_sptr self) -> pmt_vector_float

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

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

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

pc_output_buffers_full_avg(fft_vfc_sptr self) -> pmt_vector_float

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

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

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

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

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

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

fft_vfc_sptr.**set_window**(*fft_vfc_sptr self*, *pmt_vector_float window*) → bool

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

gnuradio.fft.**goertzel_fc**(*int rate*, *int len*, *float freq*) → goertzel_fc_sptr

Goertzel single-bin DFT calculation.

Constructor Specific Documentation:

Parameters:
- **rate** –
- **len** –
- **freq** –

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

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

declare_sample_delay(goertzel_fc_sptr self, unsigned int delay)

goertzel_fc_sptr.**freq**(*goertzel_fc_sptr self*) → float

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

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

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

pc_input_buffers_full_avg(goertzel_fc_sptr self) -> pmt_vector_float

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

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

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

pc_output_buffers_full_avg(goertzel_fc_sptr self) -> pmt_vector_float

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

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

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

goertzel_fc_sptr.**rate**(*goertzel_fc_sptr self*) → int

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

goertzel_fc_sptr.**set_freq**(*goertzel_fc_sptr self*, *float freq*)

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

goertzel_fc_sptr.**set_rate**(*goertzel_fc_sptr self*, *int rate*)

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

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