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

# gnuradio.video_sdl

Blocks and utilities for Video SDL module

gnuradio.video_sdl.**sink_s**(*double framerate*, *int width*, *int height*, *unsigned int format*, *int dst_width*, *int dst_height*) → sink_s_sptr

video sink using SDL

input signature is one, two or three streams of signed short. One stream: stream is grey (Y) two streems: first is grey (Y), second is alternating U and V Three streams: first is grey (Y), second is U, third is V Input samples must be in the range [0,255].

Constructor Specific Documentation:

**Parameters:**
- **framerate** –
- **width** –
- **height** –
- **format** –
- **dst_width** –
- **dst_height** –

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.video_sdl.**sink_uc**(*double framerate*, *int width*, *int height*, *unsigned int format*, *int dst_width*, *int dst_height*) → sink_uc_sptr

video sink using SDL

input signature is one, two or three streams of uchar. One stream: stream is grey (Y) two streems: first is grey (Y), second is alternating U and V Three streams: first is grey

(Y), second is U, third is V Input samples must be in the range [0,255].

Constructor Specific Documentation:

**Parameters:**
- **framerate** –
- **width** –
- **height** –
- **format** –
- **dst_width** –
- **dst_height** –

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

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

declare_sample_delay(sink_uc_sptr self, unsigned int delay)

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

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

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

pc_input_buffers_full_avg(sink_uc_sptr self) -> pmt_vector_float

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

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

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

pc_output_buffers_full_avg(sink_uc_sptr self) -> pmt_vector_float

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

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

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

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

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

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

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