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

gnuradio.gr

class gnuradio.gr.**top_block**(*name='top_block'*)

Top-level hierarchical block representing a flow-graph.

This is a python wrapper around the C++ implementation to allow python subclassing.

class gnuradio.gr.**basic_block**(*name*, *in_sig*, *out_sig*)

in_sig (gr.py_io_signature): input port signature

out_sig (gr.py_io_signature): output port signature

For backward compatibility, a sequence of numpy type names is also accepted as an io signature.

class gnuradio.gr.**block**(**args*, ***kwargs*)

The abstract base class for all 'terminal' processing blocks.

A signal processing flow is constructed by creating a tree of hierarchical blocks, which at any level may also contain terminal nodes that actually implement signal processing functions. This is the base class for all such leaf nodes.

Blocks have a set of input streams and output streams. The input_signature and output_signature define the number of input streams and output streams respectively, and the type of the data items in each stream.

Blocks report the number of items consumed on each input in general_work(), using consume() or consume_each().

If the same number of items is produced on each output, the block returns that number from general_work(). Otherwise, the block calls produce() for each output, then returns WORK_CALLED_PRODUCE. The input and output rates are not required to be related.

User derived blocks override two methods, forecast and general_work, to implement their signal processing behavior. forecast is called by the system scheduler to determine how many items are required on each input stream in order to produce a given number of output items.

general_work is called to perform the signal processing in the block. It reads the input items and writes the output items.

class gnuradio.gr.**sync_block**(*name*, *in_sig*, *out_sig*)

in_sig (gr.py_io_signature): input port signature

out_sig (gr.py_io_signature): output port signature

For backward compatibility, a sequence of numpy type names is also accepted as an io signature.

class gnuradio.gr.**sync_decimator**(**args*, ***kwargs*)

synchronous N:1 input to output with history

Override work to provide the signal processing implementation.

class gnuradio.gr.**sync_interpolator**(**args*, ***kwargs*)

synchronous 1:N input to output with history

Override work to provide the signal processing implementation.

class gnuradio.gr.**tagged_stream_block**(**args*, ***kwargs*)

Block that operates on PDUs in form of tagged streams

Override work to provide the signal processing implementation.

class gnuradio.gr.**hier_block2**(*name*, *input_signature*, *output_signature*)

Subclass this to create a python hierarchical block.

This is a python wrapper around the C++ hierarchical block implementation. Provides convenience functions and allows proper Python subclassing.

gnuradio.gr.**high_res_timer_now**() → gr::high_res_timer_type

Get the current time in ticks.

gnuradio.gr.**high_res_timer_now_perfmon**() → gr::high_res_timer_type

Get the current time in ticks - for performance monitoring.

gnuradio.gr.**high_res_timer_epoch**() → gr::high_res_timer_type

Get the tick count at the epoch.

gnuradio.gr.**high_res_timer_tps**() → gr::high_res_timer_type

Get the number of ticks per second.

gnuradio.gr.**io_signature**

alias of **make**

gnuradio.gr.**io_signature2**

alias of **make2**

gnuradio.gr.**io_signature3**

alias of **make3**

gnuradio.gr.**io_signaturev**

alias of **makev**

gnuradio.gr.**prefix**() → std::string const

return SYSCONFDIR. Typically ${CMAKE_INSTALL_PREFIX}/etc or /etc

gnuradio.gr.**prefsdir**() → std::string const

return preferences file directory. Typically ${SYSCONFDIR}/etc/conf.d

gnuradio.gr.**sysconfdir**() → std::string const

return SYSCONFDIR. Typically ${CMAKE_INSTALL_PREFIX}/etc or /etc

gnuradio.gr.**version**() → std::string const

return version string defined by cmake (GrVersion.cmake)

gnuradio.gr.**major_version**() → std::string const

return just the major version defined by cmake

gnuradio.gr.**api_version**() → std::string const

return just the api version defined by cmake

gnuradio.gr.**minor_version**() → std::string const

returnjust the minor version defined by cmake

gnuradio.gr.**prefs**

alias of **singleton**

class gnuradio.gr.**logger**(*logger_name*)

Proxy of C++ gr::logger class.

gnuradio.gr.**logger_config**(*std::string const config_filename*, *unsigned int watch_period=0*)

gnuradio.gr.**logger_get_names**() → std::vector< std::string,std::allocator< std::string > >

gnuradio.gr.**logger_reset_config**()

class gnuradio.gr.**tag_t**(**args*)

Proxy of C++ gr::tag_t class.

gnuradio.gr.**tag_t_offset_compare**(*tag_t x*, *tag_t y*) → bool

gnuradio.gr.**tag_t_offset_compare_key**()

Convert a tag_t_offset_compare function into a key=function This method is modeled after functools.cmp_to_key(_func_). It can be used by functions that accept a key function, such as sorted(), min(), max(), etc. to compare tags by their offsets, e.g., sorted(tag_list, key=gr.tag_t_offset_compare_key()).

gnuradio.gr.**tag_to_pmt**(*tag*)

Convert a Python-readable object to a stream tag

gnuradio.gr.**tag_to_python**(*tag*)

Convert a stream tag to a Python-readable object

gnuradio.gr.**tag_utils**

alias of gnuradio.gr.tag_utils

gnuradio.gr.**sizeof_gr_complex**

gnuradio.gr.**sizeof_float**

gnuradio.gr.**sizeof_int**

gnuradio.gr.**sizeof_short**

gnuradio.gr.**sizeof_char**

gnuradio.gr.**sizeof_double**

gnuradio.gr.**branchless_binary_slicer**(*float x*) → unsigned int

gnuradio.gr.**binary_slicer**(*float x*) → unsigned int

gnuradio.gr.**branchless_clip**(*float x*, *float clip*) → float

gnuradio.gr.**clip**(*float x*, *float clip*) → float

gnuradio.gr.**branchless_quad_0deg_slicer**(*float r*, *float i*) → unsigned int

branchless_quad_0deg_slicer(gr_complex x) -> unsigned int

gnuradio.gr.**quad_0deg_slicer**(*float r*, *float i*) → unsigned int

quad_0deg_slicer(gr_complex x) -> unsigned int

gnuradio.gr.**branchless_quad_45deg_slicer**(*float r*, *float i*) → unsigned int

branchless_quad_45deg_slicer(gr_complex x) -> unsigned int

gnuradio.gr.**quad_45deg_slicer**(*float r*, *float i*) → unsigned int

quad_45deg_slicer(gr_complex x) -> unsigned int

class gnuradio.gr.**feval**

Proxy of C++ gr::py_feval class.

class gnuradio.gr.**feval_cc**

Proxy of C++ gr::py_feval_cc class.

class gnuradio.gr.**feval_dd**

Proxy of C++ gr::py_feval_dd class.

class gnuradio.gr.**feval_ll**

Proxy of C++ gr::py_feval_ll class.

class gnuradio.gr.**feval_p**

Proxy of C++ gr::py_feval_p class.

gnuradio.gr.**gateway**

alias of **gnuradio.gr.gateway**