# gnuradio

GNU Radio is a free & open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in hobbyist, academic and commercial environments to support both wireless communications research and real-world radio systems.

GNU Radio applications are primarily written using the Python programming language, while the supplied performance-critical signal-processing path is implemented in C++ using processor floating-point extensions, where available. Thus, the developer is able to implement real-time, high-throughput radio systems in a simple-to-use, rapid-application-development environment.

While not primarily a simulation tool, GNU Radio does support development of signal processing algorithms using pre-recorded or generated data, avoiding the need for actual RF hardware.



Polymorphic Types.

The type can really be used to store anything, but also has simple conversion methods for common data types such as bool, long, or a vector.

The polymorphic type simplifies message passing between blocks, as all of the data is of the same type, including the message. Tags also use PMTs as data type, so a stream tag can be of any logical data type. In a sense, PMTs are a way to extend C++' strict typing with something more flexible.

The PMT library supports the following major types: bool, symbol (string), integer, real, complex, null, pair, list, vector, dict, uniform_vector, any (boost::any cast)

## Runtime

| | |
|---|---|
| gnuradio.gr.top_block | Top-level hierarchical block representing a flow-graph. |
| gnuradio.gr.basic_block | in_sig (gr.py_io_signature): input port signature |
| gnuradio.gr.block | The abstract base class for all 'terminal' processing blocks. |
| gnuradio.gr.sync_block | in_sig (gr.py_io_signature): input port signature |
| gnuradio.gr.sync_decimator | synchronous N:1 input to output with history |
| gnuradio.gr.sync_interpolator | synchronous 1:N input to output with history |
| gnuradio.gr.tagged_stream_block | Block that operates on PDUs in form of tagged streams |
| gnuradio.gr.hier_block2 | Subclass this to create a python hierarchical block. |
| gnuradio.gr.high_res_timer_now | Get the current time in ticks. |
| gnuradio.gr.high_res_timer_now_perfmon | Get the current time in ticks - for performance monitoring. |
| gnuradio.gr.high_res_timer_epoch | Get the tick count at the epoch. |
| gnuradio.gr.high_res_timer_tps | Get the number of ticks per second. |
| gnuradio.gr.io_signature | i/o signature for input and output ports. |
| gnuradio.gr.io_signature2 | Create an i/o signature. |
| gnuradio.gr.io_signature3 | Create an i/o signature. |
| gnuradio.gr.io_signaturev | Create an i/o signature. |
| gnuradio.gr.prefix | return SYSCONFDIR. Typically ${CMAKE_INSTALL_PREFIX}/etc or /etc |
| gnuradio.gr.prefsdir | return preferences file directory. Typically ${SYSCONFDIR}/etc/conf.d |
| gnuradio.gr.sysconfdir | return SYSCONFDIR. Typically ${CMAKE_INSTALL_PREFIX}/etc or /etc |
| gnuradio.gr.version | return version string defined by cmake (GrVersion.cmake) |
| gnuradio.gr.major_version | return just the major version defined by cmake |
| gnuradio.gr.api_version | return just the api version defined by cmake |
| gnuradio.gr.minor_version | returnjust the minor version defined by cmake |
| gnuradio.gr.prefs | |
| gnuradio.gr.logger | Proxy of C++ gr::logger class. |
| gnuradio.gr.logger_config | |
| gnuradio.gr.logger_get_names | |
| gnuradio.gr.logger_reset_config | |
| gnuradio.gr.tag_t | Proxy of C++ gr::tag_t class. |
| gnuradio.gr.tag_t_offset_compare | |

| | |
|---|---|
| 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_). |
| gnuradio.gr.tag_to_pmt | Convert a Python-readable object to a stream tag |
| gnuradio.gr.tag_to_python | Convert a stream tag to a Python-readable object |
| gnuradio.gr.tag_utils | |
| gnuradio.gr.sizeof_gr_complex | int(x, base=10) -> int or long |
| gnuradio.gr.sizeof_float | int(x, base=10) -> int or long |
| gnuradio.gr.sizeof_int | int(x, base=10) -> int or long |
| gnuradio.gr.sizeof_short | int(x, base=10) -> int or long |
| gnuradio.gr.sizeof_char | int(x, base=10) -> int or long |
| gnuradio.gr.sizeof_double | int(x, base=10) -> int or long |
| gnuradio.gr.branchless_binary_slicer | |
| gnuradio.gr.binary_slicer | |
| gnuradio.gr.branchless_clip | |
| gnuradio.gr.clip | |
| gnuradio.gr.branchless_quad_0deg_slicer | branchless_quad_0deg_slicer(gr_complex x) -> unsigned int |
| gnuradio.gr.quad_0deg_slicer | quad_0deg_slicer(gr_complex x) -> unsigned int |
| gnuradio.gr.branchless_quad_45deg_slicer | branchless_quad_45deg_slicer(gr_complex x) -> unsigned int |
| gnuradio.gr.quad_45deg_slicer | quad_45deg_slicer(gr_complex x) -> unsigned int |
| gnuradio.gr.feval | Proxy of C++ gr::py_feval class. |
| gnuradio.gr.feval_cc | Proxy of C++ gr::py_feval_cc class. |
| gnuradio.gr.feval_dd | Proxy of C++ gr::py_feval_dd class. |
| gnuradio.gr.feval_ll | Proxy of C++ gr::py_feval_ll class. |
| gnuradio.gr.feval_p | Proxy of C++ gr::py_feval_p class. |
| gnuradio.gr.gateway | |

## PMT

| | |
|---|---|
| pmt.acons | (acons x y a) == (cons (cons x y) a) |
| pmt.any_ref | Return underlying boost::any. |
| pmt.any_set | Store in . |
| pmt.assoc | Find the first pair in whose car field is and return that pair. |
| pmt.assq | Find the first pair in whose car field is and return that pair. |
| pmt.assv | Find the first pair in whose car field is and return that pair. |
| pmt.blob_data | Return a pointer to the blob's data. |
| pmt.blob_length | Return the blob's length in bytes. |
| pmt.c32vector_elements | |
| pmt.c32vector_ref | |
| pmt.c32vector_set | |
| pmt.c64vector_elements | |
| pmt.c64vector_ref | |
| pmt.c64vector_set | |
| pmt.caar | |
| pmt.cadddr | |
| pmt.caddr | |
| pmt.cadr | |
| pmt.car | If is a pair, return the car of the , otherwise raise wrong_type. |
| pmt.cdar | |
| pmt.cddr | |
| pmt.cdr | If is a pair, return the cdr of the , otherwise raise wrong_type. |
| pmt.cons | Return a newly allocated pair whose car is and whose cdr is . |
| pmt.deserialize | Create obj from portable byte-serial representation. |
| pmt.deserialize_str | Provide a simple string generating interface to pmt's deserialize function. |
| pmt.dict_add | Return a new dictionary with associated with . |
| pmt.dict_delete | Return a new dictionary with removed. |
| pmt.dict_has_key | Return true if exists in . |
| pmt.dict_items | Return list of (key . |
| pmt.dict_keys | Return list of keys. |
| pmt.dict_ref | If exists in , return associated value; otherwise return . |
| pmt.dict_update | Return a new dictionary with k=>v pairs from added. |
| pmt.dict_values | Return list of values. |
| pmt.dump_sizeof | |
| pmt.eq | Return true if x and y are the same object; otherwise return false. |

| | |
|---|---|
| pmt.equal | pmt::equal recursively compares the contents of pairs and vectors, applying pmt::eqv on other objects such as numbers and symbols. pmt::equal may fail to terminate if its arguments are circular data structures. |
| pmt.eqv | Return true if x and y should normally be regarded as the same object, else false. |
| pmt.f32vector_elements | |
| pmt.f32vector_ref | |
| pmt.f32vector_set | |
| pmt.f64vector_elements | |
| pmt.f64vector_ref | |
| pmt.f64vector_set | |
| pmt.from_bool | Return #f is val is false, else return #t. |
| pmt.from_complex | Return a complex number constructed of the given real and imaginary parts. |
| pmt.from_double | Return the pmt value that represents double . |
| pmt.from_float | |
| pmt.from_long | Return the pmt value that represents the integer . |
| pmt.from_uint64 | Return the pmt value that represents the uint64 . |
| pmt.get_PMT_EOF | |
| pmt.get_PMT_F | |
| pmt.get_PMT_NIL | |
| pmt.get_PMT_T | |
| pmt.init_c32vector | |
| pmt.init_c64vector | |
| pmt.init_f32vector | |
| pmt.init_f64vector | |
| pmt.init_s16vector | |
| pmt.init_s32vector | |
| pmt.init_s8vector | |
| pmt.init_u16vector | |
| pmt.init_u32vector | |
| pmt.init_u8vector | |
| pmt.intern | Alias for pmt_string_to_symbol. |
| pmt.is_any | Return true if is an any. |
| pmt.is_blob | Return true if is a blob, otherwise false. |
| pmt.is_bool | Return true if obj is #t or #f, else return false. |
| pmt.is_c32vector | |
| pmt.is_c64vector | |
| pmt.is_complex | return true if is a complex number, false otherwise. |
| pmt.is_dict | Return true if is a dictionary (warning: also returns true for a pair) |
| pmt.is_eof_object | return true if obj is the EOF object, otherwise return false. |
| pmt.is_f32vector | |
| pmt.is_f64vector | |
| pmt.is_false | Return true if obj is #f, else return true. |
| pmt.is_integer | Return true if is an integer number, else false. |
| pmt.is_msg_accepter | Return true if is a msg_accepter. |
| pmt.is_null | Return true if is the empty list, otherwise return false. |
| pmt.is_number | Return true if obj is any kind of number, else false. |
| pmt.is_pair | Return true if is a pair, else false (warning: also returns true for a dict) |
| pmt.is_real | |
| pmt.is_s16vector | |
| pmt.is_s32vector | |
| pmt.is_s64vector | |
| pmt.is_s8vector | |
| pmt.is_symbol | Return true if obj is a symbol, else false. |
| pmt.is_true | Return false if obj is #f, else return true. |
| pmt.is_tuple | Return true if is a tuple, otherwise false. |
| pmt.is_u16vector | |
| pmt.is_u32vector | |
| pmt.is_u64vector | |
| pmt.is_u8vector | |
| pmt.is_uint64 | Return true if is an uint64 number, else false. |
| pmt.is_uniform_vector | true if is any kind of uniform numeric vector |
| pmt.is_vector | Return true if is a vector, otherwise false. |
| pmt.length | Return the number of elements in v. |
| pmt.list1 | Return a list of length 1 containing . |
| pmt.list2 | Return a list of length 2 containing , . |
| pmt.list3 | Return a list of length 3 containing , , . |

| | |
|---|---|
| pmt.list4 | Return a list of length 4 containing , , , . |
| pmt.list5 | Return a list of length 5 containing , , , , . |
| pmt.list6 | Return a list of length 6 containing , , , , , . |
| pmt.list_add | Return with added to it. |
| pmt.list_has | Return bool of contains . |
| pmt.list_rm | Return with removed from it. |
| pmt.make_any | make an any |
| pmt.make_blob | Make a blob given a pointer and length in bytes. |
| pmt.make_c32vector | |
| pmt.make_c64vector | |
| pmt.make_dict | Make an empty dictionary. |
| pmt.make_f32vector | |
| pmt.make_f64vector | |
| pmt.make_msg_accepter | make a msg_accepter |
| pmt.make_rectangular | Return a complex number constructed of the given real and imaginary parts. |
| pmt.make_s16vector | |
| pmt.make_s32vector | |
| pmt.make_s64vector | |
| pmt.make_s8vector | |
| pmt.make_tuple | make_tuple(swig_int_ptr e0) -> swig_int_ptr |
| pmt.make_u16vector | |
| pmt.make_u32vector | |
| pmt.make_u64vector | |
| pmt.make_u8vector | |
| pmt.make_vector | Make a vector of length , with initial values set to . |
| pmt.map | Apply element-wise to the elements of list and returns a list of the results, in order. |
| pmt.member | Return the first sublist of whose car is . |
| pmt.memq | Return the first sublist of whose car is . |
| pmt.memv | Return the first sublist of whose car is . |
| pmt.msg_accepter_ref | Return underlying msg_accepter. |
| pmt.nth | locates element of |
| pmt.nthcdr | returns the tail of that would be obtained by calling cdr times in succession. |
| pmt.pmt_vector_cdouble | Proxy of C++ std::vector<(std::complex<(double)>)> class. |
| pmt.pmt_vector_cfloat | Proxy of C++ std::vector<(std::complex<(float)>)> class. |
| pmt.pmt_vector_double | Proxy of C++ std::vector<(double)> class. |
| pmt.pmt_vector_float | Proxy of C++ std::vector<(float)> class. |
| pmt.pmt_vector_int16 | Proxy of C++ std::vector<(int16_t)> class. |
| pmt.pmt_vector_int32 | Proxy of C++ std::vector<(int32_t)> class. |
| pmt.pmt_vector_int8 | Proxy of C++ std::vector<(int8_t)> class. |
| pmt.pmt_vector_uint16 | Proxy of C++ std::vector<(uint16_t)> class. |
| pmt.pmt_vector_uint32 | Proxy of C++ std::vector<(uint32_t)> class. |
| pmt.pmt_vector_uint8 | Proxy of C++ std::vector<(uint8_t)> class. |
| pmt.read | read converts external representations of pmt objects into the objects themselves. Read returns the next object parsable from the given input port, updating port to point to the first character past the end of the external representation of the object. |
| pmt.reverse | reverse . |
| pmt.reverse_x | destructively reverse . |
| pmt.s16vector_elements | |
| pmt.s16vector_ref | |
| pmt.s16vector_set | |
| pmt.s32vector_elements | |
| pmt.s32vector_ref | |
| pmt.s32vector_set | |
| pmt.s64vector_ref | |
| pmt.s64vector_set | |
| pmt.s8vector_elements | |
| pmt.s8vector_ref | |
| pmt.s8vector_set | |
| pmt.serialize | Write portable byte-serial representation of to . |
| pmt.serialize_str | Provide a simple string generating interface to pmt's serialize function. |
| pmt.set_car | Stores in the car field of . |
| pmt.set_cdr | Stores in the cdr field of . |
| pmt.string_to_symbol | Return the symbol whose name is . |
| pmt.subsetp | Return true if every element of appears in , and false otherwise. |

| | |
|---|---|
| pmt.symbol_to_string | If a symbol, return the name of the symbol as a string. |
| pmt.to_bool | Return true if val is pmt::True, return false when val is pmt::PMT_F,. |
| pmt.to_complex | If is complex, real or integer, return the closest complex<double>. |
| pmt.to_double | Convert pmt to double if possible. |
| pmt.to_float | Convert pmt to float if possible. |
| pmt.to_long | Convert pmt to long if possible. |
| pmt.to_pmt | |
| pmt.to_python | |
| pmt.to_tuple | If is a vector or proper list, return a tuple containing the elements of x |
| pmt.to_uint64 | Convert pmt to uint64 if possible. |
| pmt.tuple_ref | Return the contents of position of . |
| pmt.u16vector_elements | |
| pmt.u16vector_ref | |
| pmt.u16vector_set | |
| pmt.u32vector_elements | |
| pmt.u32vector_ref | |
| pmt.u32vector_set | |
| pmt.u64vector_ref | |
| pmt.u64vector_set | |
| pmt.u8vector_elements | |
| pmt.u8vector_ref | |
| pmt.u8vector_set | |
| pmt.uniform_vector_elements | |
| pmt.uniform_vector_itemsize | item size in bytes if is any kind of uniform numeric vector |
| pmt.vector_fill | Store in every position of . |
| pmt.vector_ref | Return the contents of position of . |
| pmt.vector_set | Store in position . |
| pmt.write | Write a written representation of to the given . |
| pmt.write_string | Return a string representation of . |

## Audio Signals

| | |
|---|---|
| gnuradio.audio.sink | Creates a sink from an audio device. |
| gnuradio.audio.source | Creates a source from an audio device. |
| gnuradio.vocoder.alaw_decode_bs | This block performs alaw audio decoding. |
| gnuradio.vocoder.alaw_encode_sb | This block performs g.711 alaw audio encoding. |
| gnuradio.vocoder.codec2_decode_ps | CODEC2 Vocoder Decoder |
| gnuradio.vocoder.codec2_encode_sp | CODEC2 Vocoder Encoder |
| gnuradio.vocoder.cvsd_decode_bs | This block performs CVSD audio decoding. |
| gnuradio.vocoder.cvsd_encode_sb | This block performs CVSD audio encoding. |
| gnuradio.vocoder.g721_decode_bs | This block performs g721 audio decoding. |
| gnuradio.vocoder.g721_encode_sb | This block performs g721 audio encoding. |
| gnuradio.vocoder.g723_24_decode_bs | This block performs g723_24 audio decoding. |
| gnuradio.vocoder.g723_24_encode_sb | This block performs g723_24 audio encoding. |
| gnuradio.vocoder.g723_40_decode_bs | This block performs g723_40 audio decoding. |
| gnuradio.vocoder.g723_40_encode_sb | This block performs g723_40 audio encoding. |
| gnuradio.vocoder.gsm_fr_decode_ps | GSM 06.10 Full Rate Vocoder Decoder |
| gnuradio.vocoder.gsm_fr_encode_sp | GSM 06.10 Full Rate Vocoder Encoder |
| gnuradio.vocoder.ulaw_decode_bs | This block performs ulaw audio decoding. |
| gnuradio.vocoder.ulaw_encode_sb | This block performs g.711 ulaw audio encoding. |
| gnuradio.blocks.wavfile_sink | Write stream to a Microsoft PCM (.wav) file. |
| gnuradio.blocks.wavfile_source | Read stream from a Microsoft PCM (.wav) file, output floats. |

## Boolean Operators

| | |
|---|---|
| gnuradio.blocks.and_bb | output = input[0] & input[1] & ... & input[M-1] |
| gnuradio.blocks.and_const_bb | output[m] = input[m] & value for all M streams. |
| gnuradio.blocks.and_const_ii | output[m] = input[m] & value for all M streams. |
| gnuradio.blocks.and_const_ss | output[m] = input[m] & value for all M streams. |
| gnuradio.blocks.and_ii | output = input[0] & input[1] & ... & input[M-1] |
| gnuradio.blocks.and_ss | output = input[0] & input[1] & ... & input[M-1] |
| gnuradio.blocks.not_bb | output = ~input |
| gnuradio.blocks.not_ii | output = ~input |
| gnuradio.blocks.not_ss | output = ~input |
| gnuradio.blocks.or_bb | output = input_0 \| input_1 \| , ... \| input_N) |
| gnuradio.blocks.or_ii | output = input_0 \| input_1 \| , ... \| input_N) |

| | |
|---|---|
| gnuradio.blocks.or_ss | output = input_0 \| input_1 \| , ... \| input_N) |
| gnuradio.blocks.xor_bb | output = input_0 ^ input_1 ^ , ... ^ input_N) |
| gnuradio.blocks.xor_ii | output = input_0 ^ input_1 ^ , ... ^ input_N) |
| gnuradio.blocks.xor_ss | output = input_0 ^ input_1 ^ , ... ^ input_N) |

## Byte Operators

| | |
|---|---|
| gnuradio.blocks.packed_to_unpacked_bb | Convert a stream of packed bytes or shorts to stream of unpacked bytes or shorts. |
| gnuradio.blocks.packed_to_unpacked_ii | Convert a stream of packed bytes or shorts to stream of unpacked bytes or shorts. |
| gnuradio.blocks.packed_to_unpacked_ss | Convert a stream of packed bytes or shorts to stream of unpacked bytes or shorts. |
| gnuradio.blocks.unpacked_to_packed_bb | Convert a stream of unpacked bytes or shorts into a stream of packed bytes or shorts. |
| gnuradio.blocks.unpacked_to_packed_ii | Convert a stream of unpacked bytes or shorts into a stream of packed bytes or shorts. |
| gnuradio.blocks.unpacked_to_packed_ss | Convert a stream of unpacked bytes or shorts into a stream of packed bytes or shorts. |
| gnuradio.blocks.pack_k_bits_bb | Converts a stream of bytes with 1 bit in the LSB to a byte with k relevant bits. |
| gnuradio.blocks.repack_bits_bb | Repack bits from the input stream onto bits of the output stream. |
| gnuradio.blocks.unpack_k_bits_bb | Converts a byte with k relevant bits to k output bytes with 1 bit in the LSB. |

## Channelizers

| | |
|---|---|
| gnuradio.filter.freq_xlating_fir_filter_ccc | FIR filter combined with frequency translation with gr_complex input, gr_complex output and gr_complex taps. |
| gnuradio.filter.freq_xlating_fir_filter_ccf | FIR filter combined with frequency translation with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.freq_xlating_fir_filter_fcc | FIR filter combined with frequency translation with float input, gr_complex output and gr_complex taps. |
| gnuradio.filter.freq_xlating_fir_filter_fcf | FIR filter combined with frequency translation with float input, gr_complex output and float taps. |
| gnuradio.filter.freq_xlating_fir_filter_scc | FIR filter combined with frequency translation with short input, gr_complex output and gr_complex taps. |
| gnuradio.filter.freq_xlating_fir_filter_scf | FIR filter combined with frequency translation with short input, gr_complex output and float taps. |
| gnuradio.filter.pfb_channelizer_ccf | Polyphase filterbank channelizer with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.pfb_decimator_ccf | Polyphase filterbank bandpass decimator with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.pfb_interpolator_ccf | Polyphase filterbank interpolator with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.pfb_synthesizer_ccf | Polyphase synthesis filterbank with gr_complex input, gr_complex output and float taps. |

## Channel Models

| | |
|---|---|
| gnuradio.channels.channel_model | Basic channel simulator. |
| gnuradio.channels.channel_model2 | Basic channel simulator allowing time-varying frequency and timing inputs. |
| gnuradio.channels.fading_model | fading simulator |
| gnuradio.channels.selective_fading_model | fading simulator |
| gnuradio.channels.dynamic_channel_model | dynamic channel simulator |
| gnuradio.channels.cfo_model | channel simulator |
| gnuradio.channels.sro_model | Sample Rate Offset Model. |

## Coding Blocks

| | |
|---|---|
| gnuradio.digital.additive_scrambler_bb | Scramble an input stream using an LFSR. |
| gnuradio.digital.descrambler_bb | Descramber an input stream using an LFSR. |
| gnuradio.digital.scrambler_bb | Scramble an input stream using an LFSR. |

## ControlPort Blocks

| | |
|---|---|
| gnuradio.blocks.ctrlport_probe2_b | A ControlPort probe to export vectors of signals. |
| gnuradio.blocks.ctrlport_probe2_c | A ControlPort probe to export vectors of signals. |
| gnuradio.blocks.ctrlport_probe2_f | A ControlPort probe to export vectors of signals. |
| gnuradio.blocks.ctrlport_probe2_i | A ControlPort probe to export vectors of signals. |
| gnuradio.blocks.ctrlport_probe2_s | A ControlPort probe to export vectors of signals. |
| gnuradio.blocks.ctrlport_probe_c | A ControlPort probe to export vectors of signals. |
| gnuradio.fft.ctrlport_probe_psd | A ControlPort probe to export vectors of signals. |

## Debug Blocks

| | |
|---|---|
| gnuradio.blocks.message_debug | Debug block for the message passing system. |
| gnuradio.blocks.message_strobe | Send message at defined interval. |
| gnuradio.blocks.message_strobe_random | Send message at defined interval. |
| gnuradio.blocks.tag_debug | Bit bucket that prints out any tag received. |
| gnuradio.blocks.tags_strobe | Send tags at defined interval. |
| gnuradio.blocks.vector_sink_b | unsigned char sink that writes to a vector |
| gnuradio.blocks.vector_sink_c | gr_complex sink that writes to a vector |
| gnuradio.blocks.vector_sink_f | float sink that writes to a vector |
| gnuradio.blocks.vector_sink_i | int sink that writes to a vector |
| gnuradio.blocks.vector_sink_s | short sink that writes to a vector |
| gnuradio.blocks.random_pdu | Sends a random PDU at intervals. |

## DTV Blocks

| | |
|---|---|
| gnuradio.dtv.atsc_deinterleaver | ATSC deinterleave RS encoded ATSC data ( atsc_mpeg_packet_rs_encoded > atsc_mpeg_packet_rs_encoded) |
| gnuradio.dtv.atsc_depad | ATSC depad mpeg ts packets from 256 byte atsc_mpeg_packet to 188 byte char |
| gnuradio.dtv.atsc_derandomizer | ATSC “dewhiten” incoming mpeg transport stream packets |
| gnuradio.dtv.atsc_equalizer | ATSC Receiver Equalizer. |
| gnuradio.dtv.atsc_field_sync_mux | <+description of block+> |
| gnuradio.dtv.atsc_fpll | ATSC Receiver FPLL. |
| gnuradio.dtv.atsc_fs_checker | ATSC Receiver FS_CHECKER. |
| gnuradio.dtv.atsc_interleaver | <+description of block+> |
| gnuradio.dtv.atsc_pad | <+description of block+> |
| gnuradio.dtv.atsc_randomizer | <+description of block+> |
| gnuradio.dtv.atsc_rs_decoder | ATSC Receiver Reed-Solomon Decoder. |
| gnuradio.dtv.atsc_rs_encoder | <+description of block+> |
| gnuradio.dtv.atsc_sync | ATSC Receiver SYNC. |
| gnuradio.dtv.atsc_trellis_encoder | <+description of block+> |
| gnuradio.dtv.atsc_viterbi_decoder | ATSC Viterbi Decoder. |
| gnuradio.dtv.dvb_bbheader_bb | Formats MPEG-2 Transport Stream packets into FEC baseband frames and adds a 10-byte header. |
| gnuradio.dtv.dvb_bbscrambler_bb | Scrambles FEC baseband frames with a PRBS encoder. |
| gnuradio.dtv.dvb_bch_bb | Encodes a BCH ((Bose, Chaudhuri, Hocquenghem) FEC. |
| gnuradio.dtv.dvb_ldpc_bb | Encodes a LDPC (Low-Density Parity-Check) FEC. |
| gnuradio.dtv.dvbs2_interleaver_bb | Bit interleaves DVB-S2 FEC baseband frames. |
| gnuradio.dtv.dvbs2_modulator_bc | Modulates DVB-S2 frames. |
| gnuradio.dtv.dvbs2_physical_cc | Signals DVB-S2 physical layer frames. |
| gnuradio.dtv.dvbt2_cellinterleaver_cc | Cell and time interleaves QPSK/QAM modulated cells. |
| gnuradio.dtv.dvbt2_framemapper_cc | Maps T2 frames. |
| gnuradio.dtv.dvbt2_freqinterleaver_cc | Frequency interleaves a T2 frame. |
| gnuradio.dtv.dvbt2_interleaver_bb | Bit interleaves DVB-T2 FEC baseband frames. |
| gnuradio.dtv.dvbt2_miso_cc | Splits the stream for MISO (Multiple Input Single Output). |
| gnuradio.dtv.dvbt2_modulator_bc | Modulates DVB-T2 cells. |
| gnuradio.dtv.dvbt2_p1insertion_cc | Inserts a P1 symbol. |
| gnuradio.dtv.dvbt2_paprtr_cc | Peak to Average Power Ratio (PAPR) reduction. |
| gnuradio.dtv.dvbt2_pilotgenerator_cc | Adds pilots to T2 frames. |

| | |
|---|---|
| gnuradio.dtv.dvbt_bit_inner_interleaver | Bit Inner interleaver. |
| gnuradio.dtv.dvbt_convolutional_interleaver | Convolutional interleaver. |
| gnuradio.dtv.dvbt_energy_dispersal | Energy dispersal. |
| gnuradio.dtv.dvbt_inner_coder | Inner coder with Puncturing. |
| gnuradio.dtv.dvbt_map | DVB-T mapper. |
| gnuradio.dtv.dvbt_reed_solomon_enc | Reed Solomon encoder |
| gnuradio.dtv.dvbt_reference_signals | Reference signals generator. |
| gnuradio.dtv.dvbt_symbol_inner_interleaver | Symbol interleaver. |

## Equalizer Blocks

| | |
|---|---|
| gnuradio.digital.cma_equalizer_cc | Implements constant modulus adaptive filter on complex stream. |
| gnuradio.digital.lms_dd_equalizer_cc | Least-Mean-Square Decision Directed Equalizer (complex in/out) |
| gnuradio.digital.kurtotic_equalizer_cc | Implements a kurtosis-based adaptive equalizer on complex stream. |

## Error Coding Blocks

| | |
|---|---|
| gnuradio.fec.async_decoder | Creates the decoder block for use in GNU Radio flowgraphs from a given FEC API object derived from the generic_decoder class. |
| gnuradio.fec.async_encoder | Creates the encoder block for use in GNU Radio flowgraphs with async message from a given FEC API object derived from the generic_encoder class. |
| gnuradio.fec.ber_bf | BER block in FECAPI. |
| gnuradio.fec.conv_bit_corr_bb | Correlate block in FECAPI. |
| gnuradio.fec.decode_ccsds_27_fb | A rate 1/2, k=7 convolutional decoder for the CCSDS standard. |
| gnuradio.fec.decoder | General FEC decoding block that takes in a decoder variable object (derived from gr::fec::general_decoder) for use in a flowgraph. |
| gnuradio.fec.depuncture_bb | Depuncture a stream of samples. |
| gnuradio.fec.encode_ccsds_27_bb | A rate 1/2, k=7 convolutional encoder for the CCSDS standard. |
| gnuradio.fec.encoder | Creates the encoder block for use in GNU Radio flowgraphs from a given FECAPI object derived from the generic_encoder class. |
| gnuradio.fec.generic_decoder | Parent class for FECAPI objects. |
| gnuradio.fec.generic_encoder | Proxy of C++ gr::fec::generic_encoder class. |
| gnuradio.fec.puncture_bb | Puncture a stream of unpacked bits. |
| gnuradio.fec.puncture_ff | Puncture a stream of floats. |
| gnuradio.fec.tagged_decoder | General FEC decoding block that takes in a decoder variable object (derived from gr::fec::general_decoder) for use in a flowgraph. |
| gnuradio.fec.tagged_encoder | Creates the encoder block for use in GNU Radio flowgraphs from a given FECAPI object derived from the generic_encoder class. |

## FCD Blocks

| | |
|---|---|
| gnuradio.fcd.source_c | Funcube Dongle source block. |

## File Operator Blocks

| | |
|---|---|
| gnuradio.blocks.file_descriptor_sink | Write stream to file descriptor. |
| gnuradio.blocks.file_descriptor_source | Read stream from file descriptor. |
| gnuradio.blocks.file_meta_sink | Write stream to file with meta-data headers. |
| gnuradio.blocks.file_meta_source | Reads stream from file with meta-data headers. |
| gnuradio.blocks.file_sink | Write stream to file. |
| gnuradio.blocks.file_source | Read stream from file. |
| gnuradio.blocks.tagged_file_sink | A file sink that uses tags to save files. |

## Filter Blocks

| | |
|---|---|
| gnuradio.filter.dc_blocker_cc | a computationally efficient controllable DC blocker |
| gnuradio.filter.dc_blocker_ff | a computationally efficient controllable DC blocker |
| gnuradio.filter.fft_filter_ccc | Fast FFT filter with gr_complex input, gr_complex output and gr_complex taps. |

| | |
|---|---|
| gnuradio.filter.fft_filter_ccf | Fast FFT filter with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.fft_filter_fff | Fast FFT filter with float input, float output and float taps. |
| gnuradio.filter.filter_delay_fc | Filter-Delay Combination Block. |
| gnuradio.filter.filterbank_vcvcf | Filterbank with vector of gr_complex input, vector of gr_complex output and float taps. |
| gnuradio.filter.fir_filter_ccc | FIR filter with gr_complex input, gr_complex output, and gr_complex taps. |
| gnuradio.filter.fir_filter_ccf | FIR filter with gr_complex input, gr_complex output, and float taps. |
| gnuradio.filter.fir_filter_fcc | FIR filter with float input, gr_complex output, and gr_complex taps. |
| gnuradio.filter.fir_filter_fff | FIR filter with float input, float output, and float taps. |
| gnuradio.filter.fir_filter_fsf | FIR filter with float input, short output, and float taps. |
| gnuradio.filter.fir_filter_scc | FIR filter with short input, gr_complex output, and gr_complex taps. |
| gnuradio.filter.fractional_interpolator_cc | Interpolating MMSE filter with complex input, complex output. |
| gnuradio.filter.fractional_interpolator_ff | Interpolating MMSE filter with float input, float output. |
| gnuradio.filter.fractional_resampler_cc | resampling MMSE filter with complex input, complex output |
| gnuradio.filter.fractional_resampler_ff | Resampling MMSE filter with float input, float output. |
| gnuradio.filter.freq_xlating_fir_filter_ccc | FIR filter combined with frequency translation with gr_complex input, gr_complex output and gr_complex taps. |
| gnuradio.filter.freq_xlating_fir_filter_ccf | FIR filter combined with frequency translation with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.freq_xlating_fir_filter_fcc | FIR filter combined with frequency translation with float input, gr_complex output and gr_complex taps. |
| gnuradio.filter.freq_xlating_fir_filter_fcf | FIR filter combined with frequency translation with float input, gr_complex output and float taps. |
| gnuradio.filter.freq_xlating_fir_filter_scc | FIR filter combined with frequency translation with short input, gr_complex output and gr_complex taps. |
| gnuradio.filter.freq_xlating_fir_filter_scf | FIR filter combined with frequency translation with short input, gr_complex output and float taps. |
| gnuradio.filter.hilbert_fc | Hilbert transformer. |
| gnuradio.filter.iir_filter_ccc | IIR filter with complex input, complex output, and complex taps. |
| gnuradio.filter.iir_filter_ccd | IIR filter with complex input, complex output, and double taps. |
| gnuradio.filter.iir_filter_ccf | IIR filter with complex input, complex output, and float taps. |
| gnuradio.filter.iir_filter_ccz | IIR filter with complex input, complex output, and complex (double) taps. |
| gnuradio.filter.iir_filter_ffd | IIR filter with float input, float output and double taps. |
| gnuradio.filter.interp_fir_filter_ccc | Interpolating FIR filter with gr_complex input, gr_complex output and gr_complex taps. |
| gnuradio.filter.interp_fir_filter_ccf | Interpolating FIR filter with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.interp_fir_filter_fcc | Interpolating FIR filter with float input, gr_complex output and gr_complex taps. |
| gnuradio.filter.interp_fir_filter_fff | Interpolating FIR filter with float input, float output and float taps. |
| gnuradio.filter.interp_fir_filter_fsf | Interpolating FIR filter with float input, short output and float taps. |
| gnuradio.filter.interp_fir_filter_scc | Interpolating FIR filter with short input, gr_complex output and gr_complex taps. |
| gnuradio.filter.pfb_arb_resampler_ccc | Polyphase filterbank arbitrary resampler with gr_complex input, gr_complex output and gr_complex taps. |
| gnuradio.filter.pfb_arb_resampler_ccf | Polyphase filterbank arbitrary resampler with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.pfb_arb_resampler_fff | Polyphase filterbank arbitrary resampler with float input, float output and float taps. |
| gnuradio.filter.pfb_channelizer_ccf | Polyphase filterbank channelizer with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.pfb_decimator_ccf | Polyphase filterbank bandpass decimator with gr_complex input, gr_complex output and float taps. |

| | |
|---|---|
| gnuradio.filter.pfb_interpolator_ccf | Polyphase filterbank interpolator with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.pfb_synthesizer_ccf | Polyphase synthesis filterbank with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.rational_resampler_base_ccc | Rational Resampling Polyphase FIR filter with gr_complex input, gr_complex output and gr_complex taps. |
| gnuradio.filter.rational_resampler_base_ccf | Rational Resampling Polyphase FIR filter with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.rational_resampler_base_fcc | Rational Resampling Polyphase FIR filter with float input, gr_complex output and gr_complex taps. |
| gnuradio.filter.rational_resampler_base_fff | Rational Resampling Polyphase FIR filter with float input, float output and float taps. |
| gnuradio.filter.rational_resampler_base_fsf | Rational Resampling Polyphase FIR filter with float input, short output and float taps. |
| gnuradio.filter.rational_resampler_base_scc | Rational Resampling Polyphase FIR filter with short input, gr_complex output and gr_complex taps. |
| gnuradio.filter.single_pole_iir_filter_cc | single pole IIR filter with complex input, complex output |
| gnuradio.filter.single_pole_iir_filter_ff | single pole IIR filter with float input, float output |

## Fourier Analysis

| | |
|---|---|
| gnuradio.fft.fft_vcc | Compute forward or reverse FFT. |
| gnuradio.fft.fft_vfc | Compute forward or reverse FFT. |
| gnuradio.fft.goertzel_fc | Goertzel single-bin DFT calculation. |

## Impairment Model Blocks

| |
|---|
| gnuradio.channels.amp_bal |
| gnuradio.channels.conj_fs_iqcorr |
| gnuradio.channels.distortion_2_gen |
| gnuradio.channels.distortion_3_gen |
| gnuradio.channels.impairments |
| gnuradio.channels.iqbal_gen |
| gnuradio.channels.phase_bal |
| gnuradio.channels.phase_noise_gen |
| gnuradio.channels.quantizer |

## Instrumentation Blocks

| | |
|---|---|
| gnuradio.qtgui.ber_sink_b | Constructor Specific Documentation: |
| gnuradio.qtgui.const_sink_c | A graphical sink to display the IQ constellation of multiple signals. |
| gnuradio.qtgui.freq_sink_c | A graphical sink to display multiple signals in frequency. |
| gnuradio.qtgui.freq_sink_f | A graphical sink to display multiple signals in frequency. |
| gnuradio.qtgui.histogram_sink_f | A graphical sink to display a histogram. |
| gnuradio.qtgui.number_sink | A graphical sink to display numerical values of input streams. |
| gnuradio.qtgui.sink_c | A graphical sink to display freq, spec, time, and const plots. |
| gnuradio.qtgui.sink_f | A graphical sink to display freq, spec, and time. |
| gnuradio.qtgui.time_raster_sink_b | A graphical sink to display multiple signals on a time_raster plot. |
| gnuradio.qtgui.time_raster_sink_f | A graphical sink to display multiple signals on a time_raster plot. |
| gnuradio.qtgui.time_sink_c | A graphical sink to display multiple signals in time. |
| gnuradio.qtgui.time_sink_f | A graphical sink to display multiple signals in time. |
| gnuradio.qtgui.vector_sink_f | A graphical sink to display multiple vector-based signals. |
| gnuradio.qtgui.waterfall_sink_c | A graphical sink to display multiple signals on a waterfall (spectrogram) plot. |
| gnuradio.qtgui.waterfall_sink_f | A graphical sink to display multiple signals on a waterfall (spectrogram) plot. |
| gnuradio.wxgui.histo_sink_f | Histogram module. |
| gnuradio.wxgui.oscope_sink_f | Building block for python oscilloscope module. |

## Level Control Blocks

| | |
|---|---|
| gnuradio.analog.agc2_cc | high performance Automatic Gain Control class with attack and decay rates. |
| gnuradio.analog.agc2_ff | high performance Automatic Gain Control class with attack and decay rates. |

| | |
|---|---|
| gnuradio.analog.agc3_cc | high performance Automatic Gain Control class with attack and decay rates. |
| gnuradio.analog.agc_cc | high performance Automatic Gain Control class |
| gnuradio.analog.agc_ff | high performance Automatic Gain Control class |
| gnuradio.analog.ctcss_squelch_ff | gate or zero output if CTCSS tone not present |
| gnuradio.analog.feedforward_agc_cc | Non-causal AGC which computes required gain based on max absolute value over nsamples. |
| gnuradio.blocks.moving_average_cc | output is the moving sum of the last N samples, scaled by the scale factor |
| gnuradio.blocks.moving_average_ff | output is the moving sum of the last N samples, scaled by the scale factor |
| gnuradio.blocks.moving_average_ii | output is the moving sum of the last N samples, scaled by the scale factor |
| gnuradio.blocks.moving_average_ss | output is the moving sum of the last N samples, scaled by the scale factor |
| gnuradio.blocks.mute_cc | output = input or zero if muted. |
| gnuradio.blocks.mute_ff | output = input or zero if muted. |
| gnuradio.blocks.mute_ii | output = input or zero if muted. |
| gnuradio.blocks.mute_ss | output = input or zero if muted. |
| gnuradio.analog.pwr_squelch_cc | gate or zero output when input power below threshold |
| gnuradio.analog.pwr_squelch_ff | gate or zero output when input power below threshold |
| gnuradio.analog.rail_ff | clips input values to min, max |
| gnuradio.blocks.sample_and_hold_bb | sample and hold circuit |
| gnuradio.blocks.sample_and_hold_ff | sample and hold circuit |
| gnuradio.blocks.sample_and_hold_ii | sample and hold circuit |
| gnuradio.blocks.sample_and_hold_ss | sample and hold circuit |
| gnuradio.analog.simple_squelch_cc | simple squelch block based on average signal power and threshold in dB. |
| gnuradio.blocks.threshold_ff | Output a 1 or zero based on a threshold value. |

## Math Operator Blocks

| | |
|---|---|
| gnuradio.blocks.abs_ff | output[m] = abs(input[m]) for all M streams. |
| gnuradio.blocks.abs_ii | output[m] = abs(input[m]) for all M streams. |
| gnuradio.blocks.abs_ss | output[m] = abs(input[m]) for all M streams. |
| gnuradio.blocks.add_cc | output = sum(input[0], input[1], ..., input[M-1]) |
| gnuradio.blocks.add_ff | output = sum (input_0, input_1, ...) |
| gnuradio.blocks.add_ii | output = sum(input[0], input[1], ..., input[M-1]) |
| gnuradio.blocks.add_ss | output = sum(input[0], input[1], ..., input[M-1]) |
| gnuradio.blocks.add_const_bb | output = input + constant |
| gnuradio.blocks.add_const_cc | output = input + constant |
| gnuradio.blocks.add_const_ff | output = input + constant |
| gnuradio.blocks.add_const_ii | output = input + constant |
| gnuradio.blocks.add_const_ss | output = input + constant |
| gnuradio.blocks.add_const_vbb | output[m] = input[m] + constant vector for all M streams. |
| gnuradio.blocks.add_const_vcc | output[m] = input[m] + constant vector for all M streams. |
| gnuradio.blocks.add_const_vff | output[m] = input[m] + constant vector for all M streams. |
| gnuradio.blocks.add_const_vii | output[m] = input[m] + constant vector for all M streams. |
| gnuradio.blocks.add_const_vss | output[m] = input[m] + constant vector for all M streams. |
| gnuradio.blocks.argmax_fs | Compares vectors from multiple streams and determines the index in the vector and stream number where the maximum value occurred. |
| gnuradio.blocks.argmax_is | Compares vectors from multiple streams and determines the index in the vector and stream number where the maximum value occurred. |
| gnuradio.blocks.argmax_ss | Compares vectors from multiple streams and determines the index in the vector and stream number where the maximum value occurred. |
| gnuradio.blocks.conjugate_cc | output = complex conjugate of input |
| gnuradio.blocks.divide_cc | output = input[0] / input[1] / ... / input[M-1] |
| gnuradio.blocks.divide_ff | output = input[0] / input[1] / ... / input[M-1] |
| gnuradio.blocks.divide_ii | output = input[0] / input[1] / ... / input[M-1] |
| gnuradio.blocks.divide_ss | output = input[0] / input[1] / ... / input[M-1] |
| gnuradio.blocks.integrate_cc | Integrate successive samples and decimate. |
| gnuradio.blocks.integrate_ff | Integrate successive samples and decimate. |

| | |
|---|---|
| gnuradio.blocks.integrate_ii | Integrate successive samples and decimate. |
| gnuradio.blocks.integrate_ss | Integrate successive samples and decimate. |
| gnuradio.blocks.nlog10_ff | output = n*log10(input) + k |
| gnuradio.blocks.max_ff | Compares vectors from multiple streams and determines the maximum value from each vector over all streams. |
| gnuradio.blocks.max_ii | Compares vectors from multiple streams and determines the maximum value from each vector over all streams. |
| gnuradio.blocks.max_ss | Compares vectors from multiple streams and determines the maximum value from each vector over all streams. |
| gnuradio.blocks.min_ff | Compares vectors from multiple streams and determines the minimum value from each vector over all streams. |
| gnuradio.blocks.min_ii | Compares vectors from multiple streams and determines the minimum value from each vector over all streams. |
| gnuradio.blocks.min_ss | Compares vectors from multiple streams and determines the minimum value from each vector over all streams. |
| gnuradio.blocks.multiply_cc | output = prod (input_0, input_1, ...) |
| gnuradio.blocks.multiply_ff | output = prod (input_0, input_1, ...) |
| gnuradio.blocks.multiply_ii | output = prod (input_0, input_1, ...) |
| gnuradio.blocks.multiply_ss | output = prod (input_0, input_1, ...) |
| gnuradio.blocks.multiply_matrix_ff | Matrix multiplexer/multiplier: y(k) = A x(k) |
| gnuradio.blocks.multiply_conjugate_cc | Multiplies stream 0 by the complex conjugate of stream 1. |
| gnuradio.blocks.multiply_const_cc | output = input * complex constant |
| gnuradio.blocks.multiply_const_ff | output = input * real constant |
| gnuradio.blocks.multiply_const_ii | output = input * constant |
| gnuradio.blocks.multiply_const_ss | output = input * constant |
| gnuradio.blocks.multiply_const_vcc | output = input * constant vector (element-wise) |
| gnuradio.blocks.multiply_const_vff | output = input * constant vector (element-wise) |
| gnuradio.blocks.multiply_const_vii | output = input * constant vector (element-wise) |
| gnuradio.blocks.multiply_const_vss | output = input * constant vector (element-wise) |
| gnuradio.blocks.rms_cf | RMS average power. |
| gnuradio.blocks.rms_ff | RMS average power. |
| gnuradio.blocks.rotator_cc | Complex rotator. |
| gnuradio.blocks.sub_cc | output = input_0 - input_1 - ...) |
| gnuradio.blocks.sub_ff | output = input_0 - input_1 - ...) |
| gnuradio.blocks.sub_ii | output = input_0 - input_1 - ...) |
| gnuradio.blocks.sub_ss | output = input_0 - input_1 - ...) |
| gnuradio.blocks.transcendental | A block that performs various transcendental math operations. |

Measurement Tool Blocks ——————————-_

| | |
|---|---|
| gnuradio.digital.mpsk_snr_est_cc | A block for computing SNR of a signal. |
| gnuradio.digital.probe_mpsk_snr_est_c | A probe for computing SNR of a PSK signal. |
| gnuradio.digital.probe_density_b | This block maintains a running average of the input stream and makes it available as an accessor function. |
| gnuradio.blocks.probe_rate | throughput measurement |
| gnuradio.blocks.probe_signal_b | Sink that allows a sample to be grabbed from Python. |
| gnuradio.blocks.probe_signal_c | Sink that allows a sample to be grabbed from Python. |
| gnuradio.blocks.probe_signal_f | Sink that allows a sample to be grabbed from Python. |
| gnuradio.blocks.probe_signal_i | Sink that allows a sample to be grabbed from Python. |
| gnuradio.blocks.probe_signal_s | Sink that allows a sample to be grabbed from Python. |
| gnuradio.blocks.probe_signal_vb | Sink that allows a vector of samples to be grabbed from Python. |
| gnuradio.blocks.probe_signal_vc | Sink that allows a vector of samples to be grabbed from Python. |
| gnuradio.blocks.probe_signal_vf | Sink that allows a vector of samples to be grabbed from Python. |
| gnuradio.blocks.probe_signal_vi | Sink that allows a vector of samples to be grabbed from Python. |
| gnuradio.blocks.probe_signal_vs | Sink that allows a vector of samples to be grabbed from Python. |

## Message Tool Blocks

| | |
|---|---|
| gnuradio.blocks.message_burst_source | make(size_t itemsize, msg_queue_sptr msgq) -> message_burst_source_sptr |
| gnuradio.blocks.message_debug | Debug block for the message passing system. |

| | |
|---|---|
| gnuradio.blocks.message_sink | make(size_t itemsize, msg_queue_sptr msgq, bool dont_block, std::string const & lengthtagname) -> message_sink_sptr |
| gnuradio.blocks.message_source | make(size_t itemsize, msg_queue_sptr msgq) -> message_source_sptr |
| gnuradio.blocks.message_strobe | Send message at defined interval. |
| gnuradio.blocks.message_strobe_random | Send message at defined interval. |
| gnuradio.blocks.pdu_filter | Propagates only pdus containing k=>v in meta. |
| gnuradio.blocks.pdu_remove | remove key k in pdu's meta field and pass on |
| gnuradio.blocks.pdu_set | Set k=>v in pdu's meta field and pass on. |
| gnuradio.blocks.pdu_to_tagged_stream | Turns received PDUs into a tagged stream of items. |
| gnuradio.blocks.tagged_stream_multiply_length | Allows scaling of a tagged stream length tag. |
| gnuradio.blocks.tagged_stream_to_pdu | Turns received stream data and tags into PDUs and sends them through a message port. |

## Misc Blocks

| | |
|---|---|
| gnuradio.blocks.copy | output[i] = input[i] |
| gnuradio.blocks.delay | delay the input by a certain number of samples |
| gnuradio.blocks.head | copies the first N items to the output then signals done |
| gnuradio.blocks.nop | Does nothing. |
| gnuradio.blocks.null_sink | Bit bucket. |
| gnuradio.blocks.null_source | A source of zeros used mainly for testing. |
| gnuradio.blocks.skiphead | skips the first N items, from then on copies items to the output |
| gnuradio.blocks.throttle | throttle flow of samples such that the average rate does not exceed samples_per_sec. |
| gnuradio.blocks.vector_source_b | Source that streams unsigned char items based on the input vector. |
| gnuradio.blocks.vector_source_c | Source that streams gr_complex items based on the input vector. |
| gnuradio.blocks.vector_source_f | Source that streams float items based on the input vector. |
| gnuradio.blocks.vector_source_i | Source that streams int items based on the input vector. |
| gnuradio.blocks.vector_source_s | Source that streams short items based on the input vector. |

## Modulator Blocks

| | |
|---|---|
| gnuradio.analog.am_demod_cf | Generalized AM demodulation block with audio filtering. |
| gnuradio.analog.cpm | Return the taps for an interpolating FIR filter (gr::filter::interp_fir_filter_fff). |
| gnuradio.analog.cpfsk_bc | Perform continuous phase 2-level frequency shift keying modulation on an input stream of unpacked bits. |
| gnuradio.analog.frequency_modulator_fc | Frequency modulator block. |
| gnuradio.analog.fm_demod_cf | Generalized FM demodulation block with deemphasis and audio filtering. |
| gnuradio.analog.demod_20k0f3e_cf | NBFM demodulation block, 20 KHz channels |
| gnuradio.analog.demod_200kf3e_cf | WFM demodulation block, mono. |
| gnuradio.analog.fm_deemph | FM Deemphasis IIR filter. |
| gnuradio.analog.fm_preemph | FM Preemphasis IIR filter. |
| gnuradio.analog.nbfm_rx | |
| gnuradio.analog.nbfm_tx | |
| gnuradio.analog.phase_modulator_fc | Phase modulator block. |
| gnuradio.analog.quadrature_demod_cf | quadrature demodulator: complex in, float out |
| gnuradio.analog.wfm_rcv_fmdet | |
| gnuradio.analog.wfm_rcv_pll | |
| gnuradio.analog.wfm_rcv | |
| gnuradio.analog.wfm_tx | |

## Networking Tools Blocks

| | |
|---|---|
| gnuradio.blocks.socket_pdu | Creates socket interface and translates traffic to PDUs. |
| gnuradio.blocks.tcp_server_sink | Send stream trought an TCP socket. |
| gnuradio.blocks.udp_sink | Write stream to an UDP socket. |
| gnuradio.blocks.udp_source | Read stream from an UDP socket. |

## NOAA Blocks

| | |
|---|---|
| gnuradio.noaa.hrpt_decoder | NOAA HRPT Decoder. |

| | |
|---|---|
| gnuradio.noaa.hrpt_deframer | NOAA HRPT Deframer. |
| gnuradio.noaa.hrpt_pll_cf | NOAA HRPT PLL. |

## OFDM Blocks

| | |
|---|---|
| gnuradio.digital.ofdm_carrier_allocator_cvc | Create frequency domain OFDM symbols from complex values, add pilots. |
| gnuradio.digital.ofdm_chanest_vcvc | Estimate channel and coarse frequency offset for OFDM from preambles |
| gnuradio.digital.ofdm_cyclic_prefixer | Adds a cyclic prefix and performs pulse shaping on OFDM symbols. |
| gnuradio.digital.ofdm_equalizer_base | Base class for implementation details of frequency-domain OFDM equalizers. |
| gnuradio.digital.ofdm_equalizer_simpledfe | Simple decision feedback equalizer for OFDM. |
| gnuradio.digital.ofdm_equalizer_static | Very simple static equalizer for OFDM. |
| gnuradio.digital.ofdm_frame_acquisition | take a vector of complex constellation points in from an FFT and performs a correlation and equalization. |
| gnuradio.digital.ofdm_frame_equalizer_vcvc | OFDM frame equalizer. |
| gnuradio.digital.ofdm_frame_sink | Takes an OFDM symbol in, demaps it into bits of 0's and 1's, packs them into packets, and sends to to a message queue sink. |
| gnuradio.digital.ofdm_insert_preamble | insert "pre-modulated" preamble symbols before each payload. |
| gnuradio.digital.ofdm_sampler | does the rest of the OFDM stuff |
| gnuradio.digital.ofdm_serializer_vcc | make(ofdm_carrier_allocator_cvc_sptr allocator, std::string const & packet_len_tag_key, int symbols_skipped=0, std::string const & carr_offset_key, bool input_is_shifted=True) -> ofdm_serializer_vcc_sptr |
| gnuradio.digital.ofdm_sync_sc_cfb | Schmidl & Cox synchronisation for OFDM. |

## Packet Operator Blocks

| | |
|---|---|
| gnuradio.digital.crc32_async_bb | Byte-stream CRC block for async messages. |
| gnuradio.digital.crc32_bb | Byte-stream CRC block. |
| gnuradio.digital.correlate_access_code_bb | Examine input for specified access code, one bit at a time. |
| gnuradio.digital.correlate_access_code_bb_ts | Examine input for specified access code, one bit at a time. |
| gnuradio.digital.correlate_access_code_ff_ts | Examine input for specified access code, one bit at a time. |
| gnuradio.digital.correlate_access_code_tag_bb | Examine input for specified access code, one bit at a time. |
| gnuradio.digital.framer_sink_1 | Given a stream of bits and access_code flags, assemble packets. |
| gnuradio.digital.hdlc_deframer_bp | HDLC deframer which takes in unpacked bits, and outputs PDU binary blobs. |
| gnuradio.digital.hdlc_framer_pb | HDLC framer which takes in PMT binary blobs and outputs HDLC frames as unpacked bits, with CRC and bit stuffing added. |
| gnuradio.digital.header_payload_demux | Header/Payload demuxer (HPD). |
| gnuradio.digital.packet_header_default | Default header formatter for digital packet transmission. |
| gnuradio.digital.packet_headergenerator_bb | make(long header_len, std::string const & len_tag_key) -> packet_headergenerator_bb_sptr |
| gnuradio.digital.packet_sink | process received bits looking for packet sync, header, and process bits into packet |
| gnuradio.digital.simple_correlator | inverse of simple_framer (more or less) |
| gnuradio.digital.simple_framer | add sync field, seq number and command field to payload |

## Pager Blocks

| | |
|---|---|
| gnuradio.pager.flex_deinterleave | flex deinterleave description |
| gnuradio.pager.flex_frame | flex_frame. |
| gnuradio.pager.flex_parse | flex parse description |
| gnuradio.pager.flex_sync | flex sync description |

| | |
|---|---|
| gnuradio.pager.slicer_fb | slicer description |

## Peak Detector Blocks

| | |
|---|---|
| gnuradio.blocks.burst_tagger | Sets a burst on/off tag based on the value of the trigger input. |
| gnuradio.blocks.peak_detector2_fb | Detect the peak of a signal. |
| gnuradio.blocks.peak_detector_fb | Detect the peak of a signal. |
| gnuradio.blocks.peak_detector_ib | Detect the peak of a signal. |
| gnuradio.blocks.peak_detector_sb | Detect the peak of a signal. |
| gnuradio.blocks.plateau_detector_fb | Detects a plateau and marks the middle. |

## Resampler Blocks

| | |
|---|---|
| gnuradio.filter.fractional_resampler_cc | resampling MMSE filter with complex input, complex output |
| gnuradio.filter.fractional_resampler_ff | Resampling MMSE filter with float input, float output. |
| gnuradio.filter.pfb.arb_resampler_ccf | Convenience wrapper for the polyphase filterbank arbitrary resampler. |
| gnuradio.filter.pfb.arb_resampler_fff | Convenience wrapper for the polyphase filterbank arbitrary resampler. |
| gnuradio.filter.pfb.arb_resampler_ccc | Convenience wrapper for the polyphase filterbank arbitrary resampler. |
| gnuradio.filter.pfb_arb_resampler_ccc | Polyphase filterbank arbitrary resampler with gr_complex input, gr_complex output and gr_complex taps. |
| gnuradio.filter.pfb_arb_resampler_ccf | Polyphase filterbank arbitrary resampler with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.pfb_arb_resampler_fff | Polyphase filterbank arbitrary resampler with float input, float output and float taps. |
| gnuradio.filter.rational_resampler_fff | |
| gnuradio.filter.rational_resampler_ccf | |
| gnuradio.filter.rational_resampler_ccc | |
| gnuradio.filter.rational_resampler_base_ccc | Rational Resampling Polyphase FIR filter with gr_complex input, gr_complex output and gr_complex taps. |
| gnuradio.filter.rational_resampler_base_ccf | Rational Resampling Polyphase FIR filter with gr_complex input, gr_complex output and float taps. |
| gnuradio.filter.rational_resampler_base_fcc | Rational Resampling Polyphase FIR filter with float input, gr_complex output and gr_complex taps. |
| gnuradio.filter.rational_resampler_base_fff | Rational Resampling Polyphase FIR filter with float input, float output and float taps. |
| gnuradio.filter.rational_resampler_base_fsf | Rational Resampling Polyphase FIR filter with float input, short output and float taps. |
| gnuradio.filter.rational_resampler_base_scc | Rational Resampling Polyphase FIR filter with short input, gr_complex output and gr_complex taps. |

## Stream Operator Blocks

| | |
|---|---|
| gnuradio.blocks.deinterleave | deinterleave an input block of samples into N outputs. |
| gnuradio.blocks.endian_swap | Convert stream of items into their byte swapped version. |
| gnuradio.blocks.keep_m_in_n | decimate a stream, keeping the first items out of every starting after items. |
| gnuradio.blocks.keep_one_in_n | decimate a stream, keeping the last item out of every . |
| gnuradio.blocks.patterned_interleaver | Interleave items based on the provided vector . |
| gnuradio.blocks.regenerate_bb | Detect the peak of a signal and repeat every period samples. |
| gnuradio.blocks.repeat | repeat each input times |
| gnuradio.blocks.stream_mux | Stream muxing block to multiplex many streams into one with a specified format. |
| gnuradio.blocks.stream_to_streams | convert a stream of items into a N streams of items |
| gnuradio.blocks.stream_to_tagged_stream | Converts a regular stream into a tagged stream. |
| gnuradio.blocks.stream_to_vector | convert a stream of items into a stream of gnuradio/blocks containing nitems_per_block |
| gnuradio.blocks.streams_to_stream | Convert N streams of 1 item into a 1 stream of N items. |
| gnuradio.blocks.streams_to_vector | convert N streams of items to 1 stream of vector length N |

| | |
|---|---|
| gnuradio.blocks.stretch_ff | adjust y-range of an input vector by mapping to range (max-of-input, stipulated-min). Primarily for spectral signature matching by normalizing spectrum dynamic ranges. |
| gnuradio.blocks.tagged_stream_mux | Combines tagged streams. |
| gnuradio.blocks.vector_insert_b | source of unsigned char's that gets its data from a vector |
| gnuradio.blocks.vector_insert_c | source of gr_complex's that gets its data from a vector |
| gnuradio.blocks.vector_insert_f | source of float's that gets its data from a vector |
| gnuradio.blocks.vector_insert_i | source of int's that gets its data from a vector |
| gnuradio.blocks.vector_insert_s | source of short's that gets its data from a vector |
| gnuradio.blocks.vector_to_stream | convert a stream of gnuradio/blocks of nitems_per_block items into a stream of items |
| gnuradio.blocks.vector_to_streams | Convert 1 stream of vectors of length N to N streams of items. |

## Stream Tag Tool Blocks

| | |
|---|---|
| gnuradio.blocks.stream_to_tagged_stream | Converts a regular stream into a tagged stream. |
| gnuradio.blocks.tag_gate | Control tag propagation. |
| gnuradio.blocks.tagged_stream_align | Align a stream to a tagged stream item. |
| gnuradio.blocks.tagged_stream_multiply_length | Allows scaling of a tagged stream length tag. |
| gnuradio.blocks.tagged_stream_mux | Combines tagged streams. |

## Symbol Coding Blocks

| | |
|---|---|
| gnuradio.digital.binary_slicer_fb | Slice float binary symbol producing 1 bit output. |
| gnuradio.digital.chunks_to_symbols_bc | Map a stream of unpacked symbol indexes to stream of float or complex constellation points in D dimensions (D = 1 by default) |
| gnuradio.digital.chunks_to_symbols_bf | Map a stream of unpacked symbol indexes to stream of float or complex constellation points in D dimensions (D = 1 by default) |
| gnuradio.digital.chunks_to_symbols_ic | Map a stream of unpacked symbol indexes to stream of float or complex constellation points in D dimensions (D = 1 by default) |
| gnuradio.digital.chunks_to_symbols_if | Map a stream of unpacked symbol indexes to stream of float or complex constellation points in D dimensions (D = 1 by default) |
| gnuradio.digital.chunks_to_symbols_sc | Map a stream of unpacked symbol indexes to stream of float or complex constellation points in D dimensions (D = 1 by default) |
| gnuradio.digital.chunks_to_symbols_sf | Map a stream of unpacked symbol indexes to stream of float or complex constellation points in D dimensions (D = 1 by default) |
| gnuradio.digital.constellation_decoder_cb | Constellation Decoder. |
| gnuradio.digital.constellation_soft_decoder_cf | Constellation Decoder. |
| gnuradio.digital.diff_decoder_bb | Differential encoder: y[0] = (x[0] - x[-1]) % M. |
| gnuradio.digital.diff_encoder_bb | Differential decoder: y[0] = (x[0] + y[-1]) % M. |
| gnuradio.digital.diff_phasor_cc | Differential decoding based on phase change. |
| gnuradio.digital.map_bb | output[i] = map[input[i]] |

## Synchronizer Blocks

| | |
|---|---|
| gnuradio.digital.clock_recovery_mm_cc | Mueller and M?ller (M&M) based clock recovery block with complex input, complex output. |
| gnuradio.digital.clock_recovery_mm_ff | Mueller and M?ller (M&M) based clock recovery block with float input, float output. |
| gnuradio.digital.correlate_and_sync_cc | Correlate to a preamble and send time/phase sync info. |
| gnuradio.digital.corr_est_cc | Correlate stream with a pre-defined sequence and estimate peak. |
| gnuradio.digital.costas_loop_cc | A Costas loop carrier recovery module. |
| gnuradio.digital.fll_band_edge_cc | Frequency Lock Loop using band-edge filters. |
| gnuradio.digital.mpsk_receiver_cc | This block takes care of receiving M-PSK modulated signals through phase, frequency, and symbol synchronization. |
| gnuradio.digital.msk_timing_recovery_cc | MSK/GMSK timing recovery |

| | |
|---|---|
| gnuradio.analog.pll_carriertracking_cc | Implements a PLL which locks to the input frequency and outputs the input signal mixed with that carrier. |
| gnuradio.analog.pll_freqdet_cf | Implements a PLL which locks to the input frequency and outputs an estimate of that frequency. |
| gnuradio.analog.pll_refout_cc | Implements a PLL which locks to the input frequency and outputs a carrier. |
| gnuradio.digital.pn_correlator_cc | PN code sequential search correlator. |
| gnuradio.digital.pfb_clock_sync_ccf | Timing synchronizer using polyphase filterbanks. |
| gnuradio.digital.pfb_clock_sync_fff | Timing synchronizer using polyphase filterbanks. |

## Trellis Coding Blocks

| | |
|---|---|
| gnuradio.trellis.constellation_metrics_cf | Evaluate metrics for use by the Viterbi algorithm. |
| gnuradio.trellis.encoder_bb | make(fsm FSM, int ST, int K) -> encoder_bb_sptr |
| gnuradio.trellis.encoder_bi | make(fsm FSM, int ST, int K) -> encoder_bi_sptr |
| gnuradio.trellis.encoder_bs | make(fsm FSM, int ST, int K) -> encoder_bs_sptr |
| gnuradio.trellis.encoder_ii | make(fsm FSM, int ST, int K) -> encoder_ii_sptr |
| gnuradio.trellis.encoder_si | make(fsm FSM, int ST, int K) -> encoder_si_sptr |
| gnuradio.trellis.encoder_ss | make(fsm FSM, int ST, int K) -> encoder_ss_sptr |
| gnuradio.trellis.metrics_c | Evaluate metrics for use by the Viterbi algorithm. |
| gnuradio.trellis.metrics_f | Evaluate metrics for use by the Viterbi algorithm. |
| gnuradio.trellis.metrics_i | Evaluate metrics for use by the Viterbi algorithm. |
| gnuradio.trellis.metrics_s | Evaluate metrics for use by the Viterbi algorithm. |
| gnuradio.trellis.pccc_decoder_b | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_decoder_combined_cb | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_decoder_combined_ci | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_decoder_combined_cs | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_decoder_combined_fb | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_decoder_combined_fi | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_decoder_combined_fs | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_decoder_i | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_decoder_s | Constructor Specific Documentation: |
| gnuradio.trellis.pccc_encoder_bb | PCCC encoder. |
| gnuradio.trellis.pccc_encoder_bi | PCCC encoder. |
| gnuradio.trellis.pccc_encoder_bs | PCCC encoder. |
| gnuradio.trellis.pccc_encoder_ii | PCCC encoder. |
| gnuradio.trellis.pccc_encoder_si | PCCC encoder. |
| gnuradio.trellis.pccc_encoder_ss | PCCC encoder. |
| gnuradio.trellis.permutation | Permutation. |
| gnuradio.trellis.sccc_decoder_b | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_decoder_combined_cb | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_decoder_combined_ci | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_decoder_combined_cs | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_decoder_combined_fb | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_decoder_combined_fi | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_decoder_combined_fs | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_decoder_i | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_decoder_s | Constructor Specific Documentation: |
| gnuradio.trellis.sccc_encoder_bb | SCCC encoder. |
| gnuradio.trellis.sccc_encoder_bi | SCCC encoder. |
| gnuradio.trellis.sccc_encoder_bs | SCCC encoder. |
| gnuradio.trellis.sccc_encoder_ii | SCCC encoder. |
| gnuradio.trellis.sccc_encoder_si | SCCC encoder. |
| gnuradio.trellis.sccc_encoder_ss | SCCC encoder. |
| gnuradio.trellis.siso_combined_f | Constructor Specific Documentation: |
| gnuradio.trellis.siso_f | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_b | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_cb | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_ci | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_cs | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_fb | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_fi | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_fs | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_ib | Constructor Specific Documentation: |

| | |
|---|---|
| gnuradio.trellis.viterbi_combined_ii | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_is | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_sb | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_si | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_combined_ss | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_i | Constructor Specific Documentation: |
| gnuradio.trellis.viterbi_s | Constructor Specific Documentation: |

## Type Converter Blocks

| | |
|---|---|
| gnuradio.blocks.char_to_float | Convert stream of chars to a stream of float. |
| gnuradio.blocks.char_to_short | Convert stream of chars to a stream of shorts. |
| gnuradio.blocks.complex_to_arg | complex in, arg (arctan) out (float) |
| gnuradio.blocks.complex_to_float | Convert a stream of gr_complex to 1 or 2 streams of float. |
| gnuradio.blocks.complex_to_imag | Produces the imaginary part (as a float0 of a complex stream. |
| gnuradio.blocks.complex_to_interleaved_short | Convert stream of complex to a stream of interleaved shorts. |
| gnuradio.blocks.complex_to_mag | complex in, magnitude out (float) |
| gnuradio.blocks.complex_to_mag_squared | complex in, magnitude squared out (float) |
| gnuradio.blocks.complex_to_real | Produces the real part (as a float0 of a complex stream. |
| gnuradio.blocks.float_to_char | Convert stream of floats to a stream of char. |
| gnuradio.blocks.float_to_complex | one or two floats in, complex out |
| gnuradio.blocks.float_to_int | Convert stream of floats to a stream of ints. |
| gnuradio.blocks.float_to_short | Convert stream of floats to a stream of shorts. |
| gnuradio.blocks.float_to_uchar | Convert stream of floats to a stream of unsigned chars. |
| gnuradio.blocks.int_to_float | Convert stream of ints to a stream of floats. |
| gnuradio.blocks.interleaved_char_to_complex | Convert stream of interleaved chars to a stream of complex. |
| gnuradio.blocks.interleaved_short_to_complex | Convert stream of interleaved shorts to a stream of complex. |
| gnuradio.blocks.short_to_char | Convert stream of shorts to a stream of chars. |
| gnuradio.blocks.short_to_float | Convert stream of shorts to a stream of floats. |
| gnuradio.blocks.uchar_to_float | Convert stream of unsigned chars to a stream of floats. |

## UHD Blocks

| |
|---|
| gnuradio.uhd.amsg_source |
| gnuradio.uhd.usrp_sink |
| gnuradio.uhd.usrp_source |

## Video Blocks

| | |
|---|---|
| gnuradio.video_sdl.sink_s | video sink using SDL |
| gnuradio.video_sdl.sink_uc | video sink using SDL |

## Waveform Generator Blocks

| | |
|---|---|
| gnuradio.analog.fastnoise_source_c | gnuradio.analog.fastnoise_source_f |
| gnuradio.analog.fastnoise_source_i | gnuradio.analog.fastnoise_source_s |
| gnuradio.analog.noise_source_c | gnuradio.analog.noise_source_f |
| gnuradio.analog.noise_source_i | gnuradio.analog.noise_source_s |
| gnuradio.digital.glfsr_source_b | gnuradio.digital.glfsr_source_f |
| gnuradio.analog.sig_source_c | gnuradio.analog.sig_source_f |
| gnuradio.analog.sig_source_i | gnuradio.analog.sig_source_s |

## ZeroMQ Interface Blocks

| | |
|---|---|
| gnuradio.zeromq.pub_msg_sink | Sink the contents of a msg port to a ZMQ PUB socket. |
| gnuradio.zeromq.pub_sink | Sink the contents of a stream to a ZMQ PUB socket. |
| gnuradio.zeromq.pull_msg_source | Receive messages on ZMQ PULL socket and output async messages. |

| | |
|---|---|
| gnuradio.zeromq.pull_source | Receive messages on ZMQ PULL socket and source stream. |
| gnuradio.zeromq.push_msg_sink | Sink the contents of a msg port to a ZMQ PUSH socket. |
| gnuradio.zeromq.push_sink | Sink the contents of a stream to a ZMQ PUSH socket. |
| gnuradio.zeromq.rep_msg_sink | Sink the contents of a msg port to a ZMQ REP socket. |
| gnuradio.zeromq.rep_sink | Sink the contents of a stream to a ZMQ REP socket. |
| gnuradio.zeromq.req_msg_source | Receive messages on ZMQ REQ socket output async messages. |
| gnuradio.zeromq.req_source | Receive messages on ZMQ REQ socket and source stream. |
| gnuradio.zeromq.sub_msg_source | Receive messages on ZMQ SUB socket and output async messages. |
| gnuradio.zeromq.sub_source | Receive messages on ZMQ SUB socket and source stream. |

## Helper Classes: Analog

| | |
|---|---|
| gnuradio.analog.cpm | Return the taps for an interpolating FIR filter (gr::filter::interp_fir_filter_fff). |
| gnuradio.analog.squelch_base_cc | basic squelch block; to be subclassed for other squelches. |
| gnuradio.analog.squelch_base_ff | basic squelch block; to be subclassed for other squelches. |
| gnuradio.analog.cpm | Return the taps for an interpolating FIR filter (gr::filter::interp_fir_filter_fff). |
| gnuradio.analog.squelch_base_cc | basic squelch block; to be subclassed for other squelches. |
| gnuradio.analog.squelch_base_ff | basic squelch block; to be subclassed for other squelches. |
| gnuradio.analog.am_demod_cf | Generalized AM demodulation block with audio filtering. |
| gnuradio.analog.demod_10k0a3e_cf | AM demodulation block, 10 KHz channel. |
| gnuradio.analog.fm_demod_cf | Generalized FM demodulation block with deemphasis and audio filtering. |
| gnuradio.analog.demod_20k0f3e_cf | NBFM demodulation block, 20 KHz channels |
| gnuradio.analog.demod_200kf3e_cf | WFM demodulation block, mono. |
| gnuradio.analog.fm_deemph | FM Deemphasis IIR filter. |
| gnuradio.analog.fm_preemph | FM Preemphasis IIR filter. |
| gnuradio.analog.nbfm_rx | |
| gnuradio.analog.nbfm_tx | |
| gnuradio.analog.ctcss_gen_f | |
| gnuradio.analog.standard_squelch | |
| gnuradio.analog.wfm_rcv_fmdet | |
| gnuradio.analog.wfm_rcv_pll | |
| gnuradio.analog.wfm_rcv | |
| gnuradio.analog.wfm_tx | |

## Helper Classes: Digital

| | |
|---|---|
| gnuradio.digital.constellation | An abstracted constellation object. |
| gnuradio.digital.lfsr | Fibonacci Linear Feedback Shift Register using specified polynomial mask. |
| gnuradio.digital.mpsk_snr_est | A parent class for SNR estimators, specifically for M-PSK signals in AWGN channels. |
| gnuradio.digital.simple_framer | add sync field, seq number and command field to payload |
| gnuradio.digital.crc32 | crc32(std::string const buf) -> unsigned int |
| gnuradio.digital.update_crc32 | update_crc32(unsigned int crc, std::string const buf) -> unsigned int |
| gnuradio.digital.bpsk_mod | Hierarchical block for RRC-filtered BPSK modulation. |
| gnuradio.digital.bpsk_demod | Hierarchical block for RRC-filtered BPSK demodulation. |

| | |
|---|---|
| gnuradio.digital.dbpsk_mod | Hierarchical block for RRC-filtered DBPSK modulation. |
| gnuradio.digital.dbpsk_demod | Hierarchical block for RRC-filtered DBPSK demodulation. |
| gnuradio.digital.constellation_map_generator | Uses the a basis constellation provided (e.g., from psk_constellation.psk_4()) and the the k and permutation index (pi) to generate a new Gray-coded symbol map to the constellation points provided in the basis. |
| gnuradio.digital.cpm_mod | Hierarchical block for Continuous Phase modulation. |
| gnuradio.digital.gen_and_append_crc32 | |
| gnuradio.digital.check_crc32 | |
| gnuradio.digital.generic_mod | Hierarchical block for RRC-filtered differential generic modulation. |
| gnuradio.digital.generic_demod | Hierarchical block for RRC-filtered differential generic demodulation. |
| gnuradio.digital.gfsk_mod | |
| gnuradio.digital.gfsk_demod | |
| gnuradio.digital.gmsk_mod | Hierarchical block for Gaussian Minimum Shift Key (GMSK) modulation. |
| gnuradio.digital.gmsk_demod | Hierarchical block for Gaussian Minimum Shift Key (GMSK) demodulation. |
| gnuradio.digital.type_1_mods | |
| gnuradio.digital.add_type_1_mod | |
| gnuradio.digital.type_1_demods | |
| gnuradio.digital.add_type_1_demod | |
| gnuradio.digital.type_1_constellations | |
| gnuradio.digital.add_type_1_constellation | |
| gnuradio.digital.extract_kwargs_from_options | Given a function, a list of excluded arguments and the result of parsing command line options, create a dictionary of key word arguments suitable for passing to the function. |
| gnuradio.digital.extract_kwargs_from_options_for_class | Given command line options, create dictionary suitable for passing to __init__ |
| gnuradio.digital.ofdm_packet_utils.conv_packed_binary_string_to_1_0_string | '¯' –> '10101111' |
| gnuradio.digital.ofdm_packet_utils.conv_1_0_string_to_packed_binary_string | '10101111' -> ('¯', False) |
| gnuradio.digital.ofdm_packet_utils.is_1_0_string | |
| gnuradio.digital.ofdm_packet_utils.string_to_hex_list | |
| gnuradio.digital.ofdm_packet_utils.whiten | |
| gnuradio.digital.ofdm_packet_utils.dewhiten | |
| gnuradio.digital.ofdm_packet_utils.make_header | |
| gnuradio.digital.ofdm_packet_utils.make_packet | Build a packet, given access code, payload, and whitener offset |
| gnuradio.digital.ofdm_packet_utils.unmake_packet | Return (ok, payload) |
| gnuradio.digital.ofdm_mod | Modulates an OFDM stream. |
| gnuradio.digital.ofdm_demod | Demodulates a received OFDM stream. |
| gnuradio.digital.ofdm_receiver | Performs receiver synchronization on OFDM symbols. |
| gnuradio.digital.ofdm_sync_fixed | |

| | |
|---|---|
| gnuradio.digital.ofdm_sync_ml | |
| gnuradio.digital.ofdm_sync_pnac | |
| gnuradio.digital.ofdm_sync_pn | |
| gnuradio.digital.ofdm_tx | Hierarchical block for OFDM modulation. |
| gnuradio.digital.ofdm_rx | Hierarchical block for OFDM demodulation. |
| gnuradio.digital.packet_utils.conv_packed_binary_string_to_1_0_string | '¯' –> '10101111' |
| gnuradio.digital.packet_utils.conv_1_0_string_to_packed_binary_string | '10101111' -> ('¯', False) |
| gnuradio.digital.packet_utils.is_1_0_string | |
| gnuradio.digital.packet_utils.string_to_hex_list | |
| gnuradio.digital.packet_utils.whiten | |
| gnuradio.digital.packet_utils.dewhiten | |
| gnuradio.digital.packet_utils.make_header | |
| gnuradio.digital.packet_utils.make_packet | Build a packet, given access code, payload, and whitener offset |
| gnuradio.digital.packet_utils.unmake_packet | Return (ok, payload) |
| gnuradio.digital.mod_pkts | Wrap an arbitrary digital modulator in our packet handling framework. |
| gnuradio.digital.demod_pkts | Wrap an arbitrary digital demodulator in our packet handling framework. |
| gnuradio.digital.psk_2_0x0 | 0 \| 1 |
| gnuradio.digital.psk_2_0x1 | 1 \| 0 |
| gnuradio.digital.sd_psk_2_0x0 | 0 \| 1 |
| gnuradio.digital.sd_psk_2_0x1 | 1 \| 0 |
| gnuradio.digital.psk_4_0x0_0_1 | 10 \| 11 |
| gnuradio.digital.psk_4_0x1_0_1 | 11 \| 10 |
| gnuradio.digital.psk_4_0x2_0_1 | 00 \| 01 |
| gnuradio.digital.psk_4_0x3_0_1 | 01 \| 00 |
| gnuradio.digital.psk_4_0x0_1_0 | 01 \| 11 |
| gnuradio.digital.psk_4_0x1_1_0 | 00 \| 10 |
| gnuradio.digital.psk_4_0x2_1_0 | 11 \| 01 |
| gnuradio.digital.psk_4_0x3_1_0 | 10 \| 00 |
| gnuradio.digital.sd_psk_4_0x0_0_1 | 10 \| 11 |
| gnuradio.digital.sd_psk_4_0x1_0_1 | 11 \| 10 |
| gnuradio.digital.sd_psk_4_0x2_0_1 | 00 \| 01 |
| gnuradio.digital.sd_psk_4_0x3_0_1 | 01 \| 00 |
| gnuradio.digital.sd_psk_4_0x0_1_0 | 01 \| 11 |
| gnuradio.digital.sd_psk_4_0x1_1_0 | 00 \| 10 |
| gnuradio.digital.sd_psk_4_0x2_1_0 | 11 \| 01 |

| | |
|---|---|
| gnuradio.digital.sd_psk_4_0x3_1_0 | 10 \| 00 |
| gnuradio.digital.psk_constellation | Creates a PSK constellation object. |
| gnuradio.digital.psk_mod | Hierarchical block for RRC-filtered PSK modulation. |
| gnuradio.digital.psk_demod | Hierarchical block for RRC-filtered PSK modulation. |
| gnuradio.digital.qam_16_0x0_0_1_2_3 | 0010 0110 \| 1110 1010 |
| gnuradio.digital.qam_16_0x1_0_1_2_3 | 0011 0111 \| 1111 1011 |
| gnuradio.digital.qam_16_0x2_0_1_2_3 | 0000 0100 \| 1100 1000 |
| gnuradio.digital.qam_16_0x3_0_1_2_3 | 0001 0101 \| 1101 1001 |
| gnuradio.digital.qam_16_0x0_1_0_2_3 | 0001 0101 \| 1101 1001 |
| gnuradio.digital.qam_16_0x1_1_0_2_3 | 0000 0100 \| 1100 1000 |
| gnuradio.digital.qam_16_0x2_1_0_2_3 | 0011 0111 \| 1111 1011 |
| gnuradio.digital.qam_16_0x3_1_0_2_3 | 0010 0110 \| 1110 1010 |
| gnuradio.digital.sd_qam_16_0x0_0_1_2_3 | Soft bit LUT generator for constellation: |
| gnuradio.digital.sd_qam_16_0x1_0_1_2_3 | Soft bit LUT generator for constellation: |
| gnuradio.digital.sd_qam_16_0x2_0_1_2_3 | Soft bit LUT generator for constellation: |
| gnuradio.digital.sd_qam_16_0x3_0_1_2_3 | Soft bit LUT generator for constellation: |
| gnuradio.digital.sd_qam_16_0x0_1_0_2_3 | Soft bit LUT generator for constellation: |
| gnuradio.digital.sd_qam_16_0x1_1_0_2_3 | Soft bit LUT generator for constellation: |
| gnuradio.digital.sd_qam_16_0x2_1_0_2_3 | Soft bit LUT generator for constellation: |
| gnuradio.digital.sd_qam_16_0x3_1_0_2_3 | Soft bit LUT generator for constellation: |
| gnuradio.digital.qam32_holeinside_constellation | |
| gnuradio.digital.make_differential_constellation | Create a constellation with m possible symbols where m must be a power of 4. |
| gnuradio.digital.make_non_differential_constellation | |
| gnuradio.digital.qam_constellation | Creates a QAM constellation object. |

| | |
|---|---|
| gnuradio.digital.qam_mod | Hierarchical block for RRC-filtered QAM modulation. |
| gnuradio.digital.qam_demod | Hierarchical block for RRC-filtered QAM modulation. |
| gnuradio.digital.qpsk_constellation | Creates a QPSK constellation. |
| gnuradio.digital.qpsk_mod | Hierarchical block for RRC-filtered QPSK modulation. |
| gnuradio.digital.qpsk_demod | Hierarchical block for RRC-filtered QPSK demodulation. |
| gnuradio.digital.dqpsk_constellation | |
| gnuradio.digital.dqpsk_mod | Hierarchical block for RRC-filtered DQPSK modulation. |
| gnuradio.digital.dqpsk_demod | Hierarchical block for RRC-filtered DQPSK demodulation. |
| gnuradio.digital.soft_dec_table_generator | Builds a LUT that is a list of tuples. The tuple represents the |
| gnuradio.digital.soft_dec_table | Similar in nature to soft_dec_table_generator above. |
| gnuradio.digital.calc_soft_dec_from_table | Takes in a complex sample and converts it from the coordinates (-1,-1) to (1,1) into an index value. |
| gnuradio.digital.calc_soft_dec | This function takes in any consteallation and symbol symbol set (where symbols[i] is the set of bits at constellation point constel[i] and an estimate of the noise power and produces the soft decisions for the given sample. |
| gnuradio.digital.show_table | |

## Helper Classes: FEC

| | |
|---|---|
| gnuradio.fec.cc_decoder | Convolutional Code Decoding class. |
| gnuradio.fec.cc_encoder | Convolutional Code Encoding class. |
| gnuradio.fec.ccsds_encoder | CCSDS Encoding class for convolutional encoding with rate 1/2, K=7, and polynomials [109, 79]. |
| gnuradio.fec.dummy_decoder | Dummy Decoding class. |
| gnuradio.fec.dummy_encoder | Dummy Encoding class. |
| gnuradio.fec.ldpc_decoder | Proxy of C++ gr::fec::ldpc_decoder class. |
| gnuradio.fec.ldpc_encoder | Proxy of C++ gr::fec::ldpc_encoder class. |
| gnuradio.fec.repetition_decoder | Repetition Decoding class. |
| gnuradio.fec.repetition_encoder | Repetition Encoding class. |
| gnuradio.fec.tpc_decoder | Proxy of C++ gr::fec::tpc_decoder class. |
| gnuradio.fec.tpc_encoder | Proxy of C++ gr::fec::tpc_encoder class. |
| gnuradio.fec.bercurve_generator | |
| gnuradio.fec.bitreverse | |
| gnuradio.fec.bitflip | |
| gnuradio.fec.read_bitlist | |
| gnuradio.fec.read_big_bitlist | |
| gnuradio.fec.generate_symmetries | |
| gnuradio.fec.capillary_threaded_decoder | |
| gnuradio.fec.capillary_threaded_encoder | |
| gnuradio.fec.extended_async_encoder | |
| gnuradio.fec.extended_decoder | |

| | |
|---|---|
| gnuradio.fec.extended_encoder | |
| gnuradio.fec.extended_tagged_decoder | |
| gnuradio.fec.extended_tagged_encoder | |
| gnuradio.fec.fec_test | |
| gnuradio.fec.threaded_decoder | |
| gnuradio.fec.threaded_encoder | |

## Helper Classes: FFT

| | |
|---|---|
| gnuradio.fft.window | Proxy of C++ gr::fft::window class. |

## Helper Classes: Filter

| | |
|---|---|
| gnuradio.filter.filterbank.analysis_filterbank | Uniformly modulated polyphase DFT filter bank: analysis |
| gnuradio.filter.filterbank.synthesis_filterbank | Uniformly modulated polyphase DFT filter bank: synthesis |
| gnuradio.filter.firdes | Finite Impulse Response (FIR) filter design functions. |
| gnuradio.filter.pm_remez | Parks-McClellan FIR filter design using Remez algorithm. |
| gnuradio.filter.synthesis_filterbank | Uniformly modulated polyphase DFT filter bank: synthesis |
| gnuradio.filter.analysis_filterbank | Uniformly modulated polyphase DFT filter bank: analysis |
| gnuradio.filter.freq_xlating_fft_filter_ccc | |
| gnuradio.filter.optfir.low_pass | Builds a low pass filter. |
| gnuradio.filter.optfir.band_pass | Builds a band pass filter. |
| gnuradio.filter.optfir.complex_band_pass | Builds a band pass filter with complex taps by making an LPF and |
| gnuradio.filter.optfir.band_reject | Builds a band reject filter |
| gnuradio.filter.optfir.stopband_atten_to_dev | Convert a stopband attenuation in dB to an absolute value |
| gnuradio.filter.optfir.passband_ripple_to_dev | Convert passband ripple spec expressed in dB to an absolute value |
| gnuradio.filter.optfir.remezord | FIR order estimator (lowpass, highpass, bandpass, mulitiband). |
| gnuradio.filter.optfir.lporder | FIR lowpass filter length estimator. |
| gnuradio.filter.optfir.bporder | FIR bandpass filter length estimator. |
| gnuradio.filter.pfb.channelizer_ccf | Make a Polyphase Filter channelizer (complex in, complex out, floating-point taps) |
| gnuradio.filter.pfb.interpolator_ccf | Make a Polyphase Filter interpolator (complex in, complex out, floating-point taps) |
| gnuradio.filter.pfb.decimator_ccf | Make a Polyphase Filter decimator (complex in, complex out, floating-point taps) |
| gnuradio.filter.pfb.arb_resampler_ccf | Convenience wrapper for the polyphase filterbank arbitrary resampler. |
| gnuradio.filter.pfb.arb_resampler_fff | Convenience wrapper for the polyphase filterbank arbitrary resampler. |
| gnuradio.filter.pfb.arb_resampler_ccc | Convenience wrapper for the polyphase filterbank arbitrary resampler. |
| gnuradio.filter.pfb.channelizer_hier_ccf | Make a Polyphase Filter channelizer (complex in, complex out, floating-point taps) |
| gnuradio.filter.rational_resampler_fff | |
| gnuradio.filter.rational_resampler_ccf | |
| gnuradio.filter.rational_resampler_ccc | |

## Helper Classes: Trellis

| | |
|---|---|
| gnuradio.trellis.fsm | Finite State Machine Specification class. |
| gnuradio.trellis.interleaver | INTERLEAVER class. |

## Helper Classes: UHD

| | |
|---|---|
| gnuradio.uhd.usrp_block | Base class for USRP blocks. |

## Helper Classes: Vocoder

| | |
|---|---|
| gnuradio.vocoder.codec2 | Proxy of C++ gr::vocoder::codec2 class. |
| gnuradio.vocoder.cvsd_encode_fb | This is a wrapper for the CVSD encoder that performs interpolation and filtering necessary to work with the vocoding. |
| gnuradio.vocoder.cvsd_decode_bf | This is a wrapper for the CVSD decoder that performs decimation and filtering necessary to work with the vocoding. |

## Helper Classes: WXGUI

| | |
|---|---|
| gnuradio.wxgui.oscope_sink_x | Abstract class for python oscilloscope module. |
| gnuradio.wxgui.histo_sink_f | Histogram module. |