Learning Objective 5.1: I can create a GNU Radio out-of-tree (OOT) module and add a custom Python block to it.

Learning Objective 5.2: I can implement a custom signal-processing block from scratch and integrate it into a GNU Radio flowgraph.

Show your work. Write a **Documentation** line at the end (**None** if you did not collaborate).

1. **The files of a module.** A cadet runs `gr_modtool newmod beacon` and then `gr_modtool add -t sync -l python tone_finder` in the new directory.

   (a) Name the file the cadet edits for each job: (i) the signal processing, (ii) the label and parameters GRC shows, and (iii) the unit test that `ctest` runs.

   (i) the block's `work()`: `python/beacon/tone_finder.py`
   (ii) GRC's view: `grc/beacon_tone_finder.block.yml`
   (iii) the test: `python/beacon/qa_tone_finder.py`

   (b) The cadet builds and installs the module with GNU Radio 3.10.9. `ctest` passes, but a flowgraph that uses the block stops with

   ```
   ModuleNotFoundError: No module named 'gnuradio.beacon.tone_finder'
   ```

   Explain the cause and the fix.

   On 3.10.9, `gr_modtool add` does not add `tone_finder.py` to the `gr_python_install(FILES ...)` line of `python/beacon/CMakeLists.txt`. `ctest` passes because it runs from the build directory's copy of the whole `python/` folder, but the install copies only `__init__.py`. Add `tone_finder.py` to that line, then build and install again.

   (c) After a correct install, GRC's block library still has no `tone_finder`. Give two possible causes.

   GRC was open during the install and has not reloaded its library: restart it. Or the cadet installed the module under a different prefix from the GNU Radio that GRC runs, for example without setting `CMAKE_INSTALL_PREFIX` to the radioconda environment, so the YAML file is not in that environment's `share/gnuradio/grc/blocks`.

2. **Choosing a block type.** For each block below, choose sync, decimator, interpolator, or general, and give the ratio of outputs to inputs.

   (a) A block that converts each float sample to dB.

   The block makes one output per input, so it is a sync block, 1:1.

   sync

   (b) A block that outputs the peak magnitude of each run of 100 complex samples.

   The block makes one output per 100 inputs, so it is a decimator with decim = 100.

   decimator, 100

   (c) A block that holds each bit for 8 samples.

   The block makes eight outputs per input, so it is an interpolator with interp = 8; Repeat does this job.

   interpolator, 8

   (d) A squelch that passes only the samples whose power is above a threshold and drops the rest.

   The number of outputs per input depends on the signal, so the block must report what it consumed and produced each call: general.

   general

   (e) The Energy Detector could have been a general block. Explain in words what code a decimator lets the programmer leave out.

   As a decimator, the block tells GNU Radio its fixed ratio once, and the scheduler then hands `work()` exactly $n \times N$ inputs for n outputs and consumes them itself. A general block would have to write a `forecast()` function and call `consume()` with the number of inputs used on every call.

3. **Levels.** A cadet runs the Energy Detector on a keyed tone at 48 kHz, averaging 64 samples. On the Power plot, the floor between marks sits near -12 dB and wanders about half a dB up and down, and the carrier sits just above 0 dB.

   (a) Choose a threshold halfway between the two levels on the plot.

   $$\frac{-12+0}{2} = -6 \text{ dB}$$

   about -6 dB

   (b) How long does each output cover, and how many outputs does the block make per second?

   $$T = \frac{64}{48{,}000} = 1.33 \text{ ms}$$
   $$\frac{48{,}000}{64} = 750 \text{ outputs per second}$$

   1.33 ms, 750/s

   (c) Suggest a hysteresis and explain the choice in words.

   A hysteresis of about 2 to 3 dB works. That is several times the half-dB wander of the floor, so noise near the threshold cannot turn the key back off, and the off level, -8 to -9 dB, still sits 3 dB or more above the floor.

   (d) The cadet changes Samples Averaged from 64 to 16. Describe what changes on the Power plot and in the key's timing, and whether the levels from (a) and (c) still work.

   The floor gets more ragged, since each output averages fewer samples, and its peaks reach closer to the off level. Each output covers only 0.33 ms, so the key's edges land more precisely. The threshold can stay near -6 dB; check the floor's peaks against the off level on the plot, and widen the hysteresis if they reach it.

4. **Hysteresis by hand.** The Energy Detector, with its key starting at 0, receives these ten power values in dB:

$$-25,\ -14,\ -16,\ -14,\ -16,\ -14,\ -21,\ -16,\ -14,\ -22$$

(a) Write the ten key outputs for a threshold of -15 dB and a hysteresis of 4 dB, and count the times the key changes.

The key turns on above -15 dB and off below -19 dB, so it reads `0 1 1 1 1 1 0 0 1 0`, with 4 changes.

(b) Repeat for a hysteresis of 0 dB.

The key turns on above -15 dB and off below -15 dB, so it reads `0 1 0 1 0 1 0 0 1 0`, with 8 changes.

(c) The values -14 and -16 dB are noise on a carrier that stays on. Explain in words which key is correct, and what a Morse decoder would make of the other.

The key with 4 dB of hysteresis is correct: it stays on from the first -14 dB until the power drops to -21 dB, when the carrier goes off. Without hysteresis, the key flips at every dip below -15 dB, and a decoder would read one long element as three short ones.

5. **A unit test.** A test feeds the Energy Detector, with 4 samples averaged, a threshold of -10 dB, and a hysteresis of 3 dB, these twelve complex samples:

$$1,\ 1,\ 1,\ 1,\ \ 0.1,\ 0.1,\ 0.1,\ 0.1,\ \ 1,\ 1,\ 0,\ 0$$

Averaged four at a time, these samples read 0, -20, and -3 dB on the power output.

(a) How many outputs does each port make, and which input samples does each output average?

Twelve inputs at 4 per output give 3 outputs on each port. The first averages samples 1 to 4, the second 5 to 8, and the third 9 to 12.

3 on each port

(b) Give the three key outputs.

The first output, 0 dB, is above -10 dB, so the key turns on; -20 dB is below -13 dB, so it turns off; -3 dB is above -10 dB, so it turns on again. The key is 1, 0, 1.

1, 0, 1

(c) The last output averages a run where the carrier switched off halfway through. Explain in words what that does to the timing of a key edge, and how the number of samples averaged sets the size of the effect.

The block sees only the average of each run, so an edge inside a run moves to the run's boundary, and the run reads 3 dB down if the carrier was on for half of it. The key's edges are therefore uncertain by up to one output, N/f_s seconds: averaging more samples smooths the floor but blurs the timing.

Documentation: _____