Learning Objective 6.6: I can perform RF reverse engineering on an unknown signal: identify its modulation, symbol rate, and payload structure.

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

1. **Finding the carrier.** A Simulated wireless doorbell is recorded at 1 MS/s, centered at 314.900 MHz, which is an FCC Part 15 frequency. Opened in a viewer like `L17-sigmf-view.grc`, the waterfall shows a narrow line 150 kHz below the center that stays on the whole recording, and short bursts 80 kHz above the center that appear each time the button is pressed.

   (a) Find the doorbell's carrier frequency, and say how you know which signal is the doorbell.

   The bursts appear and stop with each press, so they are the doorbell; the steady line belongs to something that is always on.

   $$f_c = 314.900 + 0.080 = 314.980 \text{ MHz}$$

314.980 MHz

   (b) Give the Frequency Xlating FIR Filter's Center Frequency that shifts the doorbell to 0 Hz.

   The filter's center frequency is the offset from the recording's center that it moves to 0 Hz: $+80{,}000$ Hz.

80000

   (c) The bursts look several kilohertz wide on the spectrum, and the steady line looks almost as thin as one FFT bin. In words, explain why.

   Switching a carrier on and off spreads its power over a band of frequencies around it, and the faster it switches, the wider the band. The steady line never changes, so its power stays at one frequency.

2. **The meta file.** A teammate sends a SigMF recording. Its `.sigmf-meta` file holds these lines:

```
"core:datatype": "ci16_le",  "core:sample_rate": 2000000.0,  "core:frequency":
                                    433000000.0
```

(a) Give the SigMF Source (Minimal) block's Output Type, the block that must follow it, and the flowgraph's `samp_rate` and `fc`.

The SigMF Source's documentation says `ci16_le` is read as short, followed by an **IShort To Complex** block, which pairs the shorts into complex samples. `samp_rate` = 2000000 and `fc` = 433e6.

(b) `ci16_le` stores each sample as two 16-bit numbers. How many bytes does a 5 s recording take, and how many would `cf32_le` take?

40 MB; 80 MB

$$5 \times 2{,}000{,}000 \times 4 \text{ bytes} = 40 \text{ MB}$$
$$5 \times 2{,}000{,}000 \times 8 \text{ bytes} = 80 \text{ MB}$$

(c) The SigMF Source (Minimal) in `L17-chips.grc` has its Output Type set to complex float (`fc32_le`). Explain in words what the flowgraph would show if you opened this `ci16_le` file without changing it.

The source would read every 8 bytes as one complex float sample, but each 8 bytes holds two `ci16` samples. The values would be meaningless, and the displays would show noise-like garbage instead of bursts, with half as many samples as the file holds. Set the Output Type to complex short (`sc16_le`) and add an IShort To Complex block and a Multiply Const of 1/32768, or convert the file to `cf32_le` first.

3. **Chips to bits.** `L17-chips.grc` opens the doorbell's recording at 1 MS/s with a chip length of 300 samples, and its Chip Reader prints every burst as the same line of chips:

```
1000100010101010001010001
```

(a) Find the chip length in microseconds.

$$\frac{300}{1{,}000{,}000 \text{ samples/s}} = 300\ \mu\text{s}$$

300 μs

(b) Every pulse is one chip long, and the gaps after the pulses are one chip or three chips. Name the encoding, and explain why the line ends with a lone pulse.

This is **pulse-distance** coding: every pulse is the same, and the length of the gap after it carries the bit. A gap is measured from one pulse to the next, so the last bit's gap needs a pulse after it to end; the lone `1` at the end closes the last gap.

(c) Take a long gap, `1000`, as a 1 and a short gap, `10`, as a 0. Decode the line into 8 bits and write them in hex.

Split the line after each pulse's gap: `1000 1000 10 10 10 1000 10 1000 1`, which is 1 1 0 0 0 1 0 1 and a final pulse: $1100\,0101_2 =$ 0xC5.

0xC5

(d) Find how long this frame lasts, and explain in words why a pulse-distance remote has no single bit rate.

The line is 25 chips: $25 \times 300\ \mu\text{s} = 7.5$ ms. A 1 takes 4 chips and a 0 takes 2, so a frame of mostly 1s takes longer than one of mostly 0s; only the chip rate, $1/300\ \mu\text{s} \approx 3333$ baud, is fixed.

4. **Encoder settings.** A Simulated garage remote uses pulse-width coding with 4 chips per bit, `1110` for a 1 and `1000` for a 0, with 350 μs chips. Each frame is 12 bits followed by 10 ms of silence. You build its encoder as the L17 lesson did, at $f_s = 1$ MS/s: a Vector Source of the bits, a Map, Unpack K Bits, Char To Float, a Repeat, and a Stream Mux with a Constant Source of 0.

   (a) Give the Map's table and Unpack K Bits' K.

   The Map replaces each bit with the number whose binary form is its chip pattern: 0 with $1000_2 = 8$ and 1 with $1110_2 = 14$. Unpack K Bits with $K = 4$ writes those 4 bits out, most significant first.

   [8, 14 ; K = 4][0cm]

   (b) Give the Repeat's interpolation and the Stream Mux's two lengths.

   350; [16800, 10000][0cm]

$$\text{Repeat} = 350\,\mu\text{s} \times 10^6 = 350$$
$$\text{frame} = 12 \times 4 \times 350 = 16{,}800$$
$$\text{gap} = 0.010 \times 10^6 = 10{,}000$$

   (c) Your encoder's output, opened in `L17-chips.grc` beside the original recording, shows the same lines of chips except that every one of yours is 4 chips longer. In words, give the most likely mistake.

   Four chips are one bit, so the encoder sends 13 bits instead of 12: the Vector Source holds one bit too many, or the Stream Mux's frame length counts 13 bits. Comparing the extra chips with the original shows which bit was added.

5. **Which modulation?** Four unknown recordings open in a viewer like `L17-sigmf-view.grc`. Name the modulation of each from what the displays show, and say which lesson's signal it resembles.

   (a) The waterfall shows two parallel tracks a few kilohertz apart, and the signal is always on one of them; the envelope is constant.

   Frequency-shift keying: the carrier moves between two frequencies, which is Lesson 8's FSK.

   (b) The waterfall shows one band that stays on and does not move, and the envelope is nearly constant; a constellation plot after carrier recovery shows four clusters.

   Phase-shift keying, here QPSK: the amplitude and frequency stay put and the phase carries the data, which is Lesson 15's mPSK.

   (c) Short bursts appear at one frequency, and the envelope switches between zero and one level in pulses of two lengths.

   On-off keying with pulse-width coding, which is the L17 fan remote, and the keyed carrier of Lessons 11 and 12.

   (d) Short dashes of the same width appear at a new frequency every few hundred milliseconds, stepping around a band.

   Frequency hopping, as in Lesson 16: one narrowband signal moving from channel to channel. A receiver tuned to one channel would see only the hops that land on its channel, so the viewer has to show the whole band.

6. **Repeats and noise.** The L17 fan remote sends each frame 6 times per press and ends each frame with a check nibble.

   (a) In words, explain what the repeats and the check each let a receiver do when the signal is weak.

   Noise can flip a chip and spoil a frame. The check lets the receiver recognize a spoiled frame and ignore it instead of acting on the wrong command, and the repeats give it five more chances to receive a frame that passes the check.

   (b) You lower the signal in the pulse decoder explorer by raising the noise. Name two things you would watch to see the frames start to fail, in the explorer or in `L17-chips.grc`.

   In the explorer, groups marked `?` that are neither `110` nor `100`, and now and then the Frame readout reporting that the check fails. In `L17-chips.grc`, the lines of chips for one press no longer matching each other from repeat to repeat.