

WIR-101 Anchor Reading Guide

3,669 words · ~17 min read

*VCA-WIR-101 cross-chapter reading-guide handout. Companion to the catalog page at <https://virtuscyberacademy.org/vca-wir-101>. Audience: Belt-3 RF/wireless-track students arriving from the catalog's distilled "What Belt-3 RF/Wireless Track Graduates Recognize" register. *

The catalog page tells you WHAT the course covers. This guide tells you HOW to read the canonical anchors that build the Belt-3: which books, which chapters, in which order, what to extract on each pass, and how the anchors compose into a coherent vocabulary that prepares you for RF-201, OSWP-track wireless-pentest credential prep, and hands-on capstone work.

§0. What this guide is for

WIR-101 is the academy's wireless-track foundations course. Four anchors carry the Belt-3: Lyons's *Understanding Digital Signal Processing* for the signal-processing vocabulary, Wyglinski et al.'s *SDR for Engineers* for the practitioner SDR-front-end vocabulary, Lichtman's PySDR for the build-it-yourself complement, and the GNU Radio tutorials and flowgraph corpus for the canonical software-radio platform. Two supplementary anchors (Smith's *DSP Guide* and the Ossmann HackRF videos) are mentioned briefly in this section but do not earn full per-anchor walks at the Belt-3.

The four primary anchors do not compose a textbook tour. Each is opinionated about what matters at this level, and the set has been chosen so that the pair Lyons + Wyglinski supplies the down-to-earth narrative vocabulary while Lichtman + GNU Radio supply the build-it-yourself complement. By the end of WIR-101 a student should be able to (a) discuss the sampling theorem, aliasing, FIR and IIR filter behaviour, and the analogue-to-digital boundary in the same vocabulary the tools and texts use, (b) build small SDR flowgraphs in GNU Radio Companion that exercise that vocabulary against real spectrum captures, and (c) move into RF-201 with the practitioner-foundation literacy that the next belt assumes.

This guide is opinionated by design. It is not a comprehensive bibliography. Anchors that other peer programs lean on (Proakis's *Digital Communications*; Pozar's *Microwave Engineering*; ARRL handbooks read cover to cover) are deliberately not the primary anchors at this level because they are encyclopedic rather than opinionated, or because they sit at a register that crowds out the SDR-first vocabulary the academy uses. WIR-101 graduates know the comprehensive material exists; they were trained on the opinionated material.

The guide reads Lyons before Wyglinski because Wyglinski's receiver-chain vocabulary builds on Lyons's signals-and-systems vocabulary; reversing the order would force students to pick up half the foundation while reading about the front-end. Lichtman before GNU Radio because PySDR's narrative scaffolds the GNU Radio Companion exercises rather than the other way around. The reading order is the academy's argument; the tools-and-texts pairing is the academy's calibration.

§1. The anchor reading register

Four anchors. Read in this order on first pass; revisit per the per-anchor walks below for capstone preparation.

Anchor 1: Lyons, *Understanding Digital Signal Processing*, Ch 1-5

Edition / pointer: Richard G. Lyons, *Understanding Digital Signal Processing*, 3rd edition (Pearson, 2010; ISBN 978-0-13-702741-5). Lyons received the IEEE SPS Educator of the Year award in 2012; the book is the canonical practitioner-tier DSP reference at this level. Chapters 1-5 cover the WIR-101 vocabulary; later chapters (advanced filter design, advanced spectral analysis) belong to RF-201 and RF-301. Library-acquire or paperback ~\$80-110.

Why this matters at Belt-3: Lyons writes for engineers learning DSP rather than for mathematicians proving theorems about it. The chapter sequence walks the discrete-time signal vocabulary from periodic sampling through the DFT through FIR and IIR filter behaviour, with a deliberate practitioner emphasis: the diagrams are tool-shaped (filter coefficients, impulse responses, frequency-domain plots), the worked examples are tractable in a Jupyter notebook, and the cross-references treat aliasing, leakage, and quantisation as practical concerns rather than as edge cases. A WIR-101 graduate who has internalised Chapters 1-5 reads any subsequent SDR text or any GNU Radio block with the right vocabulary already installed; without Lyons the same student has to learn the vocabulary while also learning the tool, which doubles the cognitive load.

Suggested reading order: First. Read Lyons before Wyglinski because Wyglinski assumes the Lyons vocabulary as foundation. Read Lyons before Lichtman because PySDR's narrative scaffolds onto Lyons's signal-and-system frame. The first-read pass should land Chapters 1-5; revisit specific chapters as labs encounter the material at depth.

Cross-link to academy artifacts: WIR-101 Lab 2 (sampling-theorem demonstration with RTL-SDR captures); WIR-101 Lab 3 (FFT-driven spectrum-plot exercise on captured signals); RF-201 Ch 4-5 (advanced filter design picks up at the post-Lyons register); ARRL Technician/General study materials (Lyons's vocabulary maps directly onto the wave-and-spectrum portions of the question pool).

Anchor 2: Wyglinski et al., *Software-Defined Radio for Engineers*, Ch 1-3

Edition / pointer: Alexander M. Wyglinski, Robin Getz, Travis F. Collins, and Di Pu, *Software-Defined Radio for Engineers*, Artech House, 2018. Free PDF via Analog Devices's developer site (the book was sponsored by ADI for the PlutoSDR community). Chapters 1-3 cover the WIR-101 entry register on RF front-ends, ADCs/DACs, and IQ representation; later chapters (advanced receiver chains, modulation deep dives) belong to RF-201 and RF-301.

Why this matters at Belt-3: Wyglinski opens with a question Lyons does not ask: where does the analogue end and the digital begin? Chapters 1-3 walk the analogue-to-digital boundary as a moving line that radio generations push toward the antenna. The student learns that a software radio's receiver is a particular budget allocation across the analogue front-end (LNA, mixer, filter, ADC) and the digital back-end (numerically-controlled oscillator, decimator, DSP). A WIR-101 graduate who has internalised Wyglinski Chapters 1-3 reads any SDR's block diagram and can name what the blocks do and why, which is the practitioner-foundation literacy RF-201 assumes.

Suggested reading order: Second. Read Wyglinski after Lyons because Wyglinski uses the Lyons vocabulary throughout. The free-PDF availability makes the read low-friction; students should plan to keep the PDF open as a reference during the WIR-101 GNU Radio labs.

Cross-link to academy artifacts: WIR-101 Lab 4 (RTL-SDR receiver-chain exercise; observe the ADC clip ceiling and the LNA noise floor); WIR-101 Lab 5 (IQ-format demonstration); RF-201 Ch 3 (advanced receiver-chain budgeting at Belt-4); RF-301 reading-guide handout (Wyglinski returns at Ch 4-5 for the engagement-tier budget framework). The WIR-101 reading is foundation; the RF-301 reading is engagement-application of the same author's deeper chapters.

Anchor 3: Lichtman, *PySDR: A Guide to SDR and DSP using Python*

Edition / pointer: Marc Lichtman, *PySDR: A Guide to SDR and DSP using Python*, free online at pysdr.org (continuously updated; cited in GNU Radio's official SuggestedReading). The full guide is browser-readable; selected chapters can be exported to PDF. No printed edition.

Why this matters at Belt-3: Lichtman's PySDR is the build-it-yourself complement to Lyons + Wyglinski. The guide walks the same DSP and SDR vocabulary as Lyons + Wyglinski but reframes every concept as a runnable Python notebook. The student does not just read about FFT-based spectrum analysis; they implement it in numpy and matplotlib and watch their own RTL-SDR capture under the algorithm. The build-it-yourself frame is what differentiates a Belt-3 graduate from a Belt-3 reader; PySDR is the academy's primary build-it-yourself anchor for the wireless track.

Suggested reading order: Third. Read PySDR after Lyons and Wyglinski because the build-it-yourself frame is most useful when the read-it-first vocabulary has been installed. Students should treat PySDR as a lab companion: open the chapter, type the code, run it against captured spectrum, compare the output to the textbook prediction.

Cross-link to academy artifacts: Pyodide-hosted academy Workbench (PySDR runs against academy-curated capture corpora in-browser via Pyodide; no install needed for the entry-tier exercises); WIR-101 Labs 2-7 (every primary lab has a PySDR companion notebook); RF-201 + RF-301 (PySDR continues as the build-it-yourself frame at deeper register; the chapters covered grow with the belt).

Anchor 4: GNU Radio Tutorials + GRC flowgraph corpus

Edition / pointer: GNU Radio Tutorials (free first-party documentation at gnuradio.org/tutorials) plus the GNU Radio Companion (GRC) flowgraph corpus shipped with the academy's WIR-101 lab harness. GNU Radio is the canonical software-radio platform; the tutorials and flowgraph corpus are the canonical first-introduce reference for the discipline.

Why this matters at Belt-3: GNU Radio is the working software-radio platform the academy and the practitioner community both run. Reading the tutorials and walking the flowgraph corpus teaches the student the language and tooling vocabulary of the discipline: blocks, taps, queues, sample rates, decimation, throttling, sinks, sources. A WIR-101 graduate who has internalised the GNU Radio vocabulary can read any community flowgraph (the gr-osmosdr corpus, the gr-iio corpus, third-party demodulator flowgraphs) and understand what it does, which is the practitioner-foundation literacy RF-201 assumes.

Suggested reading order: Fourth. Read GNU Radio's tutorials after Lichtman because PySDR's notebook frame translates more directly to GRC's flowgraph frame than the other way around (both express the same data flow at different layers of abstraction). Students should walk the GRC tutorials in order: introduction, a first flowgraph, sources and sinks, frequency-domain analysis, FM receiver, the advanced examples.

Cross-link to academy artifacts: WIR-101 Labs 6-9 (every primary GNU Radio lab walks a specific tutorial first); RF-201 capstone (the URH plus GRC pairing earns its place at Belt-4); RF-301 reading-guide handout (Mitola's cognitive-radio framing reads against the GNU Radio platform at Belt-5).

§2. Lyons deep walk: signals, sampling, and discrete-time foundations

Lyons's Chapters 1-5 carry the DSP vocabulary every subsequent anchor assumes. Read for the practitioner frame as the chapter's central operational primitive.

What to extract

A Belt-3 graduate should carry the following five facts from Lyons Ch 1-5:

1. **Periodic sampling is the foundational discretisation.** A continuous signal becomes a discrete-time sequence by sampling at a fixed rate; the sampling-theorem floor (at least twice the highest signal frequency) is the constraint that determines what the discrete sequence can faithfully represent. Belt-3 graduates should be able to state the sampling theorem in their own words and explain why it sets a floor rather than a ceiling.
2. **Aliasing is the structural consequence of under-sampling.** Frequency content above the Nyquist rate reflects (folds) into the spectrum below it; the alias appears at a deterministic frequency the student can predict from the original frequency and the sample rate. Belt-3 graduates should be able to predict an alias and explain why anti-aliasing filters live in the analogue front-end.
3. **The DFT is the discrete-time window onto the frequency domain.** The Discrete Fourier Transform converts a finite sequence of samples into a finite sequence of frequency-domain bins; the FFT is the efficient algorithm that computes it. Belt-3 graduates should be able to read a spectrum-plot output and name the bin width and the time-window length that produced it.

4. **FIR and IIR filters are the two classical filter families.** Finite-impulse-response filters have impulse responses of bounded length; infinite-impulse-response filters reuse their own outputs. The trade-off is stability and design simplicity (FIR) versus efficiency and order-for-order performance (IIR). Belt-3 graduates should be able to recognise an FIR or IIR filter from its block diagram and explain the trade-off.
5. **Quantisation is the second discretisation.** Sampling discretises time; quantisation discretises amplitude. The ADC's effective number of bits sets the quantisation noise floor; quantisation noise is the unavoidable cost of the digital representation. Belt-3 graduates should be able to compute a quantisation-noise floor for a given ADC and explain why dithering is sometimes used to soften it.

What is out-of-scope at Belt-3

Lyons's later chapters cover advanced filter design (Parks-McClellan; least-squares design; multi-rate signal processing) and advanced spectral analysis (windowing; spectral estimation; periodograms). These belong at Belt-4 and Belt-5; WIR-101 students should know the topics exist and not be expected to walk them at depth. The mathematical derivations of the DFT and the sampling theorem are useful for graduate-research register and out-of-scope at the practitioner-foundation tier.

Cross-anchor connections

Lyons's vocabulary is the foundation Wyglinski Chapters 1-3 assume; reading Lyons first makes Wyglinski legible. Lyons's worked examples translate directly to PySDR's runnable notebooks; the academy's recommended pattern is to read a Lyons chapter, then run the corresponding PySDR notebook, then implement the same operation in a GNU Radio Companion flowgraph. The three-anchor reinforcement (read, run, build) is what earns the Belt-3.

§3. Wyglinski Chapters 1-3 deep walk: SDR front-end vocabulary

Wyglinski Chapters 1-3 carry the practitioner SDR-front-end vocabulary. Read for the analogue-to-digital boundary as a moving line that radio generations push toward the antenna.

What to extract

A Belt-3 graduate should carry the following five facts from Wyglinski Ch 1-3:

1. **The analogue-to-digital boundary moves.** First-generation software radios sampled at IF; modern direct-conversion receivers sample at baseband or near-baseband; future radios may sample at RF. The boundary's position determines what the analogue front-end has to do and what the digital back-end has to handle.
2. **The receiver chain is a budget across components.** LNA noise figure plus mixer conversion loss plus ADC effective bits plus digital filter ripple together set the receiver's sensitivity and dynamic range. No single component sets the performance; the budget's total is the sum.
3. **IQ representation is the SDR's canonical complex-baseband.** The complex-valued IQ signal carries both amplitude and phase information at baseband; the real-only signal at IF or RF would require a higher sample rate to carry the same information. Belt-3 graduates should be able to read an IQ trace and explain the in-phase and quadrature components.
4. **ADCs and DACs are the boundary devices.** The ADC samples the analogue input into the digital domain; the DAC reconstructs an analogue signal from a digital sequence. Both have a sample rate, a resolution (bits), and a clip ceiling. Belt-3 graduates should be able to recognise the three parameters in a datasheet.
5. **Direct-conversion architectures dominate consumer SDR.** RTL-SDR, HackRF, BladeRF, LimeSDR, PlutoSDR, and ANT-SDR all use direct-conversion architectures with on-chip mixers and filters; the trade-offs (DC offset, IQ imbalance, image rejection) are well-understood and worth knowing at the practitioner register.

What is out-of-scope at Belt-3

Wyglinski's later chapters (advanced receiver-chain budgeting at engagement depth; modulation deep dives) belong at Belt-4 and Belt-5. The book's PlutoSDR-specific exercises are useful for hands-on lab work at WIR-101 register and are covered by the academy's lab harness. The mathematical derivations of intermodulation products, IIP3, and noise figure cascading are useful for circuit-design register and out-of-scope at WIR-101.

Cross-anchor connections

Wyglinski Chapters 1-3 land directly on PySDR's RF-front-end notebook chapters; reading both in parallel installs the front-end vocabulary at both the textbook and the build-it-yourself register. Wyglinski's IQ representation maps directly onto the GNU Radio

source blocks (the RTL-SDR source emits a complex stream by default); the student who has read Wyglinski recognises the complex-stream signature in any GNU Radio flowgraph.

§4. Lichtman PySDR deep walk: build-it-yourself signal-processing fluency

PySDR is the build-it-yourself complement to Lyons + Wyglinski. Read for the run-it-yourself vocabulary that the labs and the next-belt courses assume.

What to extract

A Belt-3 graduate should carry the following five facts from PySDR's foundational chapters:

1. **numpy plus matplotlib plus pyrtlsdr is the academy's primary signal-processing toolchain.** The build-it-yourself frame uses these libraries because they are standard, free, and well-documented; every PySDR notebook can be reproduced in any Python environment with these installed.
2. **A spectrum plot is fft of capture, plus magnitude, plus log scale, plus axis labels.** The four-step recipe is the foundational diagnostic; Belt-3 graduates should be able to produce a spectrum plot from any IQ capture in under five lines of Python.
3. **A waterfall plot is a stacked sequence of spectrum plots over time.** The waterfall reveals time-frequency structure that a single spectrum plot cannot; PySDR's waterfall recipe is the second foundational diagnostic.
4. **Filter design in PySDR uses scipy.signal.** The `scipy.signal.firwin` and `scipy.signal.iirfilter` functions take design parameters and return filter coefficients; the student then applies the filter via `scipy.signal.lfilter`. Belt-3 graduates should be able to design and apply a basic low-pass filter from a spec.
5. **Real captures replace synthetic signals as the lab progresses.** The PySDR pattern is to start with synthetic signals (generated in numpy), then move to RTL-SDR captures, then to academy-curated fixture pcap. The progression is the academy's way of building the practitioner-foundation literacy.

What is out-of-scope at Belt-3

PySDR's advanced chapters (MIMO; channel estimation; OFDM; LTE; 5G NR; specific cellular protocols) belong at Belt-4 and Belt-5. The build-it-yourself frame at WIR-101 covers foundations; the deeper PySDR chapters return at RF-201 and RF-301.

Cross-anchor connections

PySDR's signal-processing recipes are the runnable form of Lyons's textbook treatment; the two-anchor pairing (read, then run) is the academy's primary teaching method for the WIR-101 vocabulary. PySDR's IQ-stream handling lands directly on Wyglinski's IQ vocabulary; the student who has read both can move between the textbook and the notebook without translation cost. PySDR's filter-design recipes prepare the student for GNU Radio's filter-design tools (gr-filter); the GRC-side equivalents are graphical rather than code-based, but the underlying parameters are the same.

§5. GNU Radio Tutorials deep walk: the canonical software-radio platform

GNU Radio is the working software-radio platform the discipline runs on. Read the tutorials for the platform's vocabulary; walk the GRC flowgraph corpus for the working examples.

What to extract

A Belt-3 graduate should carry the following five facts from the GNU Radio tutorials:

1. **A flowgraph is a directed graph of blocks.** Sources produce samples; sinks consume them; intermediate blocks transform the stream. Belt-3 graduates should be able to read a flowgraph and predict what it does without running it.
2. **Sample rates and stream types are first-class.** Every block has an input and output sample rate; mismatch produces a runtime error. Stream types (complex, float, integer, byte) are similarly first-class; mismatch produces a wiring error in GRC. Belt-3 graduates should be able to debug a sample-rate or stream-type mismatch in a flowgraph.
3. **The OOT (Out-of-Tree) module pattern lets students extend GNU Radio.** Custom blocks can be written in Python or C++ and installed alongside the in-tree blocks. WIR-101 students do not need to write OOT modules; they should know the pattern exists for RF-201 capstone work.
4. **gr-osmosdr is the canonical hardware-abstraction layer.** Every supported SDR (RTL-SDR, HackRF, BladeRF, LimeSDR, PlutoSDR, ANT-SDR) is exposed through gr-osmosdr's source and sink blocks; the same flowgraph can target multiple hardware platforms by changing the source block's parameters. Belt-3 graduates should know that the gr-osmosdr layer exists and what it abstracts.

5. **The GNU Radio examples directory is the practitioner reference.** The flowgraphs shipped with GNU Radio cover FM reception, AM reception, spectrum analysis, basic modulation and demodulation, and several digital-signal protocols. Reading and running these examples is the academy's recommended GRC literacy path.

What is out-of-scope at Belt-3

OOT module authoring (Python and C++ block development) belongs at RF-201 capstone register. Advanced flowgraphs covering specific cellular or satellite protocols belong at RF-301 register. The GRC-internal generator code (how GRC translates a graphical flowgraph into a runnable Python script) is implementation-internal and out-of-scope at the practitioner-foundation tier.

Cross-anchor connections

GNU Radio is the platform PySDR's notebooks complement; the academy's recommended reading-and-running pattern is to walk a PySDR chapter, then build the equivalent flowgraph in GRC, then capture against the same target with both. The three artifacts (the notebook, the flowgraph, the capture) compose into the Belt-3 deliverable. GRC's filter-design tools land on Lyons's filter vocabulary; the student who has read Lyons can configure a GRC filter block from the design parameters without trial and error.

§6. Summary: how the anchors compose at Belt-3

Anchor	Role	Belt-3 deliverable it supports
Lyons Ch 1-5	DSP vocabulary foundation	Lab 2 sampling-theorem; Lab 3 FFT-driven spectrum-plot; ARRL Technician question-pool wave/spectrum portion
Wygłinski Ch 1-3	SDR front-end vocabulary	Lab 4 RTL-SDR receiver-chain; Lab 5 IQ-format; capstone receiver-budget worksheet
Lichtman PySDR	Build-it-yourself complement	Every primary lab has a PySDR companion notebook; capstone notebook artifact
GNU Radio tutorials + GRC	Canonical software-radio platform	Labs 6-9 GNU Radio walks; capstone flowgraph artifact; the practitioner-foundation literacy RF-201 assumes

The composition is opinionated by design. Lyons gives the read-it-first vocabulary; Wyglinski gives the SDR-front-end vocabulary; PySDR gives the run-it-yourself complement; GNU Radio gives the platform every subsequent anchor and lab assumes. The four anchors do not reduce to one anchor; each earns its place because the others assume its content. The Belt-3 is what the composition produces.

§7. What's next at Belt-4 and Belt-5

WIR-101 graduates carry the four-anchor foundation into RF-201 (Belt 4; advanced SDR + protocol-RE entry) and into RF-301 (Belt 5; advanced SDR + waveform-RE capstone). Two of the four anchors return at deeper register:

- **Lyons returns at RF-201 and RF-301** with the advanced chapters (multi-rate signal processing; advanced spectral analysis; adaptive filtering). The same author; the deeper material; the same practitioner frame.
- **Wyglinski returns at RF-301** at Chapter 4-5 (advanced receiver-chain budgeting at engagement depth). The RF-301 reading-guide handout ([handouts/cross-chapter-rf-301-anchor-reading-guide.md](#)) is the companion.
- **PySDR continues across all three belts**; the chapters covered grow with the belt.
- **GNU Radio continues across all three belts**; OOT module authoring and advanced flowgraphs return at RF-201 and RF-301.

The Belt-4 and Belt-5 anchors that do not appear at WIR-101 (Mitola's *Cognitive Radio Architecture*; Pozar's *Microwave Engineering*; cellular-stack-specific texts; specific protocol-RE references) belong to those belts because their assumed vocabulary is the WIR-101 foundation. Reading them too early would force the student to install vocabulary while reading about its application, which is the cognitive-load doubling the academy avoids.

The capstone work WIR-101 prepares for: a five-day simulated wireless engagement against an academy-owned RF testbed, with rubber-ducky antenna characterisation, WPA2 handshake capture and crack, sub-GHz protocol classification, and BLE GATT enumeration. Students who carry the four-anchor foundation into the capstone produce stronger artifacts than students who learn the vocabulary during the capstone.

Credential-prep parallel: the ARRL Technician credential maps onto Lyons's signal-and-spectrum vocabulary plus a wireless-regulation-and-safety vocabulary the academy covers in WIR-101 lecture material. WIR-101 graduates typically sit the ARRL Technician examination within one to three months of course completion.

§8. Cross-references

Artifact	Path	Relationship
Catalog page (the page you arrived from)	<code>/vca-wir-101</code>	Distilled register; this guide is its forward-pointer destination
Companion handout: WPA-AKA progression	<code>/handouts/cross-chapter-wireless-aka-progression.md</code>	Week-4 cross-chapter sidebar; reads against the academic-and-professional credential pathway
Companion handout: RF-301 anchor reading guide	<code>/handouts/cross-chapter-rf-301-anchor-reading-guide.md</code>	Belt-5 companion handout; Mitola plus Wyglinski plus Kurose-Ross 9e
RF-201 catalog page	<code>/vca-rf-201</code>	Belt-4 RF-track; sub-GHz protocol-RE capstone; the WIR-101 foundation feeds directly into RF-201
RF-301 catalog page	<code>/vca-rf-301</code>	Belt-5 RF-track; advanced SDR + waveform-RE capstone
NET-101 catalog page	<code>/vca-net-101</code>	Companion Belt-3 networking-track; the wireless track and networking track share Bejtlich at Belt-3 (NET) and at Belt-5 (both NET-301 and RF-301)