📘 SDR/DSP Skill Development Roadmap

Combining IQ Concepts, Python Programming, PySDR Learning, and RTL-SDR Practice

🎯 Objective

To progressively master Digital Signal Processing (DSP) and Software Defined Radio (SDR) using Python, IQ data, PySDR theoretical framework, and RTL-SDR hardware through a structured path of small to advanced exercises — supporting future research, product development, and standardization work.

📶 Phase 1: SIMPLE LEVEL — *Foundational Mastery*

🔗 Goals

- Build strong intuition about IQ data and phase modulation (QPSK).
- Understand how Python communicates with SDR hardware.
- Visualize signals and extract meaning from them.
- Get comfortable running basic RTL-SDR and DSP tasks.

📌 Module 1.1 — IQ Data Basics in Python

Task	Description	Tool/Concept
✅ Simulate I/Q Data	Use `numpy` to create sine and cosine signals	NumPy
✅ Combine to Form Complex Signal	Combine into I + jQ	Python Complex Numbers
✅ Visualize I/Q Plane	Plot points in complex plane	Matplotlib

> 🎯 Outcome: Understand what IQ really means — see it, simulate it.

📌 Module 1.2 — QPSK Symbol Mapping

Task	Description	Tool/Concept
✅ Create Bitstream	Define a bitstream manually	Python List
✅ Map to IQ Symbols	2 bits → (I, Q) using QPSK	Logic
✅ Plot Constellation	Visualize symbol locations	Matplotlib

> 🎯 Outcome: See how digital bits translate into modulated IQ symbols.

📌 Module 1.3 — RTL-SDR Hardware Setup & Test

Task	Description	Tool/Concept
✅ Install Drivers	Zadig driver installation for RTL-SDR	RTL-SDR
✅ Verify Dongle	Test recognition via `rtl_test`	Command Line
✅ Capture Samples in Python	Use `pyrtlsdr` to read live IQ	Python + PyRTLSDR

> 🎯 Outcome: Successfully communicate with SDR dongle and get real data.

📌 Module 1.4 — Live IQ Plotting

Task	Description	Tool/Concept
✅ Capture Live Data	From RTL-SDR using Python	PyRTLSDR
✅ Plot IQ Samples	In real-time or batch	Matplotlib
✅ Observe Shape	Understand stationarity, noise, and drift	IQ Interpretation

> 🎯 Outcome: Build muscle memory of IQ structure and dynamic signal behavior.

🔍 Phase Completion Milestones

Milestone	Criteria
🟩 Code and run QPSK bit → IQ mapping	
🟩 Plot QPSK constellation for test stream	
🟩 Receive live IQ from RTL-SDR	
🟩 Plot real-world IQ signal	
🟩 Document observations in project log	

📂 Folder Structure (Recommended)

```
SDR_Skills/
│
├── phase1_simple/
│   ├── 1_IQ_simulation.py
│   ├── 2_QPSK_mapping_plot.py
│   ├── 3_rtlsdr_capture_test.py
│   ├── 4_plot_live_iq.py
│   └── notes/
│       └── phase1_observations.md
```

📅 Suggested Weekly Timeline

Week	Activity
Week 1	Python IQ simulation + plotting
Week 2	QPSK bit → symbol mapping + constellation
Week 3	RTL-SDR setup + live data capture

Week	Activity
Week 4	Live plotting + milestone review

📘 Documentation Tracking Template

Each module can have a log like this (in `phase1_observations.md`):

```
### Module 1.2 — QPSK Mapping

✅ Bitstream used: 11001100
✅ Symbols mapped: (1+1j), (-1+1j), etc.
✅ Issues faced: Matplotlib scaling confusion
✅ Resolved: Used `plt.axis('equal')`
💡 Insight: Visualizing IQ makes modulation logic intuitive
```