📘 How QPSK Works: Transmission and Reception Roadmap

🎯 What Is QPSK?

QPSK (Quadrature Phase Shift Keying) is a digital modulation technique used to transmit binary data over radio waves.
It encodes **2 bits per symbol** by changing the **phase** of a carrier signal.

> 🔑 Key Idea: Phase = Direction of the signal in the IQ (complex) plane
> ❌ No amplitude variation involved — all symbols lie on a circle

🧭 End-to-End QPSK System Overview

➤ Transmitter Side (TX)

Step	Action
1️⃣	Input a digital bitstream (e.g., `10110010…`)
2️⃣	Group bits into pairs (2 bits = 1 symbol)
3️⃣	Map each 2-bit pair to an I and Q value
4️⃣	Use I/Q to modulate a carrier wave (cos/sin)
5️⃣	Transmit the resulting waveform over RF

➤ Receiver Side (RX)

Step	Action
1️⃣	Receive the RF waveform (real-valued signal)

Step	Action
2️⃣	Downconvert using a local oscillator
3️⃣	Extract I(t) and Q(t) baseband signals
4️⃣	Map I/Q values back to 2-bit symbols
5️⃣	Reconstruct the bitstream

✅ In QPSK, the **entire system is digital end-to-end**: only the **carrier wave is analog** for transmission, but the **information content is digital**.

🎨 QPSK Constellation: Symbol-to-IQ Mapping

Bits	I	Q	Phase
00	+1	+1	45°
01	−1	+1	135°
11	−1	−1	225°
10	+1	−1	315°

Each 2-bit pair is mapped to a **point in the IQ plane** — called a **constellation point**.

These points are then used to create the modulated waveform that is sent over the air.

🧪 How the Signal Is Constructed

The actual waveform transmitted over the air is:

$$x(t) = I(t) \cdot \cos(2\pi f_c t) - Q(t) \cdot \sin(2\pi f_c t)$$

- `f_c` is the carrier frequency (e.g., 100 MHz for FM band)
- I(t) and Q(t) are stepwise constant for each symbol

✅ This signal is **real-valued** and can be transmitted by any radio hardware like an SDR or commercial RF front-end.

📡 SDR Hardware Role (e.g., HackRF, RTL-SDR)

- On **transmit side** (e.g., with HackRF), your Python code generates I and Q samples → modulates carrier → goes to antenna.
- On **receive side** (e.g., with RTL-SDR), the antenna captures RF signal → SDR downconverts and extracts I and Q samples → Python reads them → decodes bits.

🧠 Key Takeaways for Mastery

- QPSK is **entirely phase-based** — amplitude remains constant
- The transmitter uses I/Q to **rotate the carrier's phase**
- The receiver uses I/Q to **recover phase and decode bits**
- You can simulate **all of this in Python**, and later transmit/receive with SDR

✅ What You've Built So Far

You now understand:

- The **logic** of QPSK end-to-end
- The **structure** of IQ data
- Why **I and Q are used** instead of just one signal
- How it all fits into an SDR system

🔜 Next: Practical QPSK Exercises in Python

Here's what's coming next:

1. **Map bits to I/Q using QPSK logic**

2. **Visualize IQ constellation**
3. **Simulate transmission waveform**
4. **Add noise and observe symbol shifting**
5. **Decode received I/Q back into bits**
6. **Scale up: filtering, interpolation, timing recovery**

Would you like to jump into **Step 1: Python bit-to-IQ mapping + constellation plot**, or would you prefer to review any part of this summary again first?

You're truly building this the right way — slow, correct, complete.