

# UART, ASYNCHRONOUS SERIAL

*UART is the simplest serial link: two wires, no shared clock, an agreed baud rate and framing. How TX/RX crossover works and what both sides must match.*

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

UART is the simplest way two chips talk over a serial line: two wires, no shared clock, and one agreed speed. One wire carries data out (TX), the other carries it in (RX), and as long as both ends are set to the same baud rate and framing, bytes cross reliably. It is how a board talks to a terminal, a GPS, or a plug-in module.

## TX, RX, AND THE CROSSOVER

Each device has a transmit pin (TX) and a receive pin (RX). You wire one device's TX to the other's RX, and back again, a crossover, so each side's output lands on the other's input. Both share a common ground. It is full-duplex: both directions can carry data at the same time.

## BAUD RATE AND FRAMING

Because there is no clock line, both ends have to be preset to the same bit rate, the baud rate, with 9600 and 115200 being common. Each byte is wrapped in a frame: the idle-high line drops for one start bit, then the data bits (usually 8), an optional parity bit, then one or two stop bits. Miss the baud or the framing and the same bits arrive as garbage.

- [SparkFun. Serial Communication \(asynchronous serial, baud rate, framing\).](#) [learn.sparkfun.com](https://learn.sparkfun.com)

$$t_{\text{bit}} = 1 / \text{baud}, \quad t_{\text{frame}} (8N1) = 10 / \text{baud}$$

The math is simple. At 115200 baud each bit lasts about 8.7 us, so a plain 8N1 byte, ten bits counting its start and stop, takes about 87 us. That figure is the real ceiling on how fast a single UART link moves data.

### DEEP DIVE · HOW FAR THE TWO CLOCKS CAN DRIFT

Since each side runs its own clock, they only have to stay in step for the length of one frame, about ten bits, and the receiver resyncs on every start bit. That is why a small speed error is tolerable: the usual budget is a few percent before the last data bit samples at the wrong moment. A wildly wrong baud, double or half the target, fails completely, while a close-but-imperfect crystal usually still works. When a UART prints garbled characters, a wrong baud rate is the first thing to check.

ONE UART FRAME: IDLE HIGH, A START BIT, THE DATA BITS, THEN A STOP BIT.

On a One Thousand Drones board the USB link presents itself to your computer as a serial port, and this same start-and-stop framing carries the boot log and your `print` output while you bring the board up.

---

**CHECKPOINT****1. What must two UART devices agree on before they can talk?**

- a. **The baud rate and framing**
- b. A shared clock line
- c. Their chip-select lines

ANSWER · A

*UART sends no clock, so both ends must be preset to the same baud rate and framing, or the bytes arrive as garbage.*

**2. How is one UART device's TX wired to the other device?**

- a. TX to TX, RX to RX
- b. **In a crossover: TX to RX both ways**
- c. Both to a single shared wire

ANSWER · B

*Each side's transmit pin drives the other's receive pin, so the wiring crosses over.*

**3. UART has no clock line, so what marks the start of each byte?**

- a. A chip-select pulse
- b. A rising clock edge
- c. **A start bit that pulls the idle line low**

ANSWER · C

*The idle-high line drops for one start bit, which tells the receiver a frame is coming.*

- Prerequisite: what is a bus?
- Next: SPI, the four-wire bus