LIBRARY · REFERENCE GUIDE

WHAT IS A BUS?

A bus is the shared wires plus the rules that let chips exchange data. Serial vs parallel, what a protocol fixes, and controller vs peripheral, in plain terms.

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

A bus is a set of shared wires plus the rules for using them, and it is how the chips on a board pass data to each other. The rules are the protocol: which wire carries what, when a bit counts as valid, and who is allowed to talk. Get the bus right and two parts that have never met can trade bytes reliably.

SERIAL OR PARALLEL?

A serial bus sends bits one at a time down a couple of wires. A parallel bus sends a whole byte at once across eight or more. Parallel moves more per clock tick, but it spends a pin on every bit and every trace has to stay length-matched. On a small board, pins and space are the scarce thing, so almost every on-board bus is serial.

- SparkFun. Serial Communication (serial vs parallel, logic levels). learn.sparkfun.com

WHAT A PROTOCOL FIXES

A protocol is the agreement both sides keep. It fixes the framing (where a byte starts and stops), the timing (how fast bits move, and whether a separate clock marks each one), and the roles (which chip drives the exchange). Two chips that follow the same protocol interoperate even when different companies built them.

CONTROLLER AND PERIPHERAL

Most buses have one part that starts each transfer, the controller, and one or more that answer, the peripherals. The controller decides when a transfer happens and, on a clocked bus, supplies the clock. Older datasheets call these two roles master and slave; the idea is the same.

DEEP DIVE · CLOCKED VS CLOCKLESS BUSES

A clocked bus carries a separate clock line, so the receiver samples each bit on a clock edge and the two sides stay in step even at high speed. SPI and I2C work this way. A clockless, or asynchronous, bus sends no clock, so both ends must be preset to the same speed and agree on the bit timing in advance. UART works this way. The clock is why a clocked bus can push more data before the two sides drift out of time with each other.

SERIAL SENDS BITS ONE AT A TIME ON FEW WIRES; PARALLEL SENDS MANY AT ONCE ON MANY WIRES.

On a One Thousand Drones board every on-board data link is serial: USB back to your computer, and a serial bus out to each sensor. That is what keeps the pin count low and the board small.

CHECKPOINT

1. Why is serial usually preferred over parallel on a small board?

a. It sends a whole byte at once

b. It uses far fewer pins and traces

c. It needs no protocol

ANSWER · B

Serial spends only a couple of wires; parallel costs a pin and a matched trace for every bit, which a small board cannot spare.

2. What is a bus protocol?

a. The speed of the fastest chip

b. A single shared ground wire

c. The agreed rules for framing, timing, and roles

ANSWER · C

The protocol is the agreement both sides keep about framing, timing, and who talks; it is what lets parts from different makers interoperate.

3. On most buses, which part starts a transfer?

a. The controller

b. The slowest peripheral

c. The pull-up resistor

ANSWER · A

The controller decides when a transfer happens; peripherals answer when addressed.

- Prerequisite: voltage, current, and resistance
- Next: UART, asynchronous serial