

DEBUGGING A BUS

When a bus does nothing, the fault is usually simple: a missing pull-up, the wrong mode, an address clash, or a swapped wire. A logic analyzer shows the exact bytes.

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

When a bus does nothing, the cause is almost always simple and physical: a missing pull-up, the wrong mode, two devices at one address, or a swapped wire. The fastest way to find it is a logic analyzer, which shows you the actual bits on the wire and turns a dead bus into a specific fault you can point at.

CHECK POWER AND GROUND FIRST

Before you suspect the protocol, confirm the boring things: both parts are powered, they share a common ground, and the wires go where you think they do. A missing ground between two boards is the single most common reason a bus is silent.

THE USUAL SUSPECTS

Match the symptom to the bus. On I2C, no pull-ups means the lines never reach a valid high, so nothing moves, and two devices at the same address answer at once and collide. On SPI, the wrong mode (CPOL/CPHA) makes every byte read wrong even though the wiring is right. On UART, a baud mismatch prints garbage, and swapped TX and RX means nothing arrives at all.

THE LOGIC ANALYZER

A logic analyzer clips onto the bus lines, records their exact high-low timing, then decodes it into bytes, addresses, and acknowledgements. It shows whether the clock is even running, whether a device answered, and where the sequence broke. An inexpensive analyzer with a [sigrok/PulseView](#) setup is enough for most on-board buses.

- [SparkFun. Using the USB Logic Analyzer with sigrok PulseView \(capturing and decoding a bus\).](#) [learn.sparkfun.com](#)

WHEN A BUS IS SILENT, IN THIS ORDER

Work from the most common cause outward, so your first probe lands on the likely fault before the rare one.

1. Confirm power and a shared ground. Both parts powered, both grounds tied together, before anything else.
2. Check the wiring against the schematic. TX to RX, SDA to SDA, no swapped pair.
3. For I2C, confirm the pull-ups are present and no two devices share an address.
4. For SPI, confirm the mode (CPOL/CPHA) matches the peripheral's datasheet.

5. For UART, confirm both ends are set to the same baud rate.
6. Clip on a logic analyzer and read the actual bits: is the clock running, and did the device answer?

A CAPTURED I2C TRACE WITH THE FAILURE ANNOTATED: AN ADDRESS SENT, AND NO ACKNOWLEDGEMENT BACK.

Bringing up a One Thousand Drones board, an analyzer on the I2C lines shows the controller sending an address and then either an acknowledgement from the sensor or silence, which tells you in a single capture whether the part is alive, mis-addressed, or missing its pull-ups.

CHECKPOINT

1. Two I2C devices answer at once. What is the likely cause?

- a. A baud-rate mismatch
- b. An address clash: two devices at the same address**
- c. A missing chip-select

ANSWER · B

I2C picks a device by address, so two parts sharing an address both respond and collide.

2. A bus between two separate boards is silent. What is the most common physical cause?

- a. No shared ground between them**
- b. Too much data
- c. The clock is too fast

ANSWER · A

Without a common ground the signals have no shared reference, and the bus does nothing; check it first.

3. SPI is wired correctly but every byte reads wrong. What is the likely cause?

- a. A missing pull-up
- b. An I2C address clash
- c. The wrong SPI mode (CPOL/CPHA)**

ANSWER · C

If the mode is wrong, the two sides sample on different clock edges and the data comes out scrambled even with correct wiring.

- Related: SPI, the four-wire bus
- Related: I2C, the two-wire bus