

I2C, THE TWO-WIRE BUS

I2C puts a whole bus of devices on two wires, SDA and SCL, each with an address. Why it needs pull-up resistors, and how it trades speed for pins.

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

I2C gets a whole bus of devices onto just two wires. A clock (SCL) and a data line (SDA) are shared by every device, and each one answers to its own address. When you have several slow sensors and few pins to spare, I2C is the bus to reach for.

TWO WIRES, MANY DEVICES

Every device hangs on the same SDA and SCL pair. The controller starts a transfer by sending the 7-bit address of the device it wants; only that device responds, and the rest stay quiet. That is how a dozen parts share one pair of wires with no extra pins.

- [NXP. UM10204 I2C-bus specification and user manual \(7-bit addressing, open-drain lines, standard and fast-mode data rates\).](#)-pololu.com

OPEN-DRAIN AND THE PULL-UPS

I2C lines are open-drain: a device can only pull a line low, never drive it high. A pull-up resistor on each line restores it to high whenever nothing is pulling it down. Leave the pull-ups off and both lines sit at an undefined level and the bus does nothing, which is the classic first I2C mistake. A 4.7 kΩ pull-up is a good starting value.

- [SparkFun. I2C \(open-drain lines, pull-up resistors, addressing\).](#) learn.sparkfun.com

DEEP DIVE · HOW FAST, AND THE TRADE AGAINST SPI

I2C runs at 100 kbit/s in standard mode and 400 kbit/s in fast mode, with faster modes defined but less common (NXP UM10204). That is well below SPI, and the open-drain lines pulled up by resistors limit how fast the signals can rise back to high. The payoff is the pin count: two wires carry the whole bus no matter how many devices you add, where SPI needs another chip-select for each one. Slow and pin-thrifty is exactly the right trade for a handful of housekeeping sensors.

ONE SDA/SCL PAIR, TWO PULL-UPS, AND MANY DEVICES, EACH REACHED BY ITS ADDRESS.

On a One Thousand Drones board a small I2C sensor shares its two lines with the rest of the bus and is reached by its address, so adding it costs no new microcontroller pins.

CHECKPOINT**1. Why does I2C need pull-up resistors?**

- a. To raise the bus speed
- b. The lines are open-drain, so a pull-up sets the idle-high level**
- c. To give each device its address

ANSWER · B*Devices can only pull the lines low; the pull-up is what returns a line to high when nothing drives it.***2. How does an I2C controller pick which device it talks to?**

- a. The device's 7-bit address sent on the bus**
- b. A chip-select line per device
- c. A different clock wire per device

ANSWER · A*The controller sends a 7-bit address, and only the matching device responds; there are no per-device select lines.***3. Compared with SPI, I2C is generally which of these?**

- a. Faster but needs more pins
- b. Faster and needs fewer pins
- c. Slower but needs fewer pins**

ANSWER · C*I2C trades speed for pins: two wires for the whole bus, at a lower data rate than SPI.*

- Prerequisite: resistors
- Related: pull-ups, pull-downs, and the idle line
- Next: SPI vs I2C vs UART