

PULL-UPS, PULL-DOWNS, AND THE IDLE LINE

A floating input reads noise; a pull-up or pull-down resistor gives a line a known default. How buses set their idle level, and how to size the resistor.

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

A digital input wired to nothing floats. With no driver, its voltage wanders with noise coupled from nearby signals, so it flickers between high and low and the chip cannot tell a 1 from a 0. A pull-up or pull-down resistor fixes that by tying the line to a known default level, and getting that default wrong is a classic reason a bus sits dead.

A FLOATING PIN IS UNDEFINED

An input pin with nothing driving it has no defined voltage, so stray coupling from nearby signals swings it around unpredictably. A weak resistor to a rail settles it: a pull-up to the supply holds the line high, a pull-down to ground holds it low, until something actively drives it the other way.

- [SparkFun. Pull-up Resistors \(floating pins, when and where to use a pull-up\).](#) [learn.sparkfun.com](#)

IDLE LEVEL BY CONVENTION

Each bus sets its resting level on purpose. I2C's open-drain lines idle high through their pull-ups. A UART line idles high and drops for the start bit. A reset or enable line carries a pull-up or pull-down so the chip powers up in a known state instead of a random one.

$$I = V_{cc} / R$$

The resistor value is a trade-off. A **10 kΩ** pull-up on a **3.3 V** rail passes only about **0.33 mA** when the line is held low, which is easy on the driver but slow to pull the line back high. A stronger **1 kΩ** snaps it high faster and costs about **3.3 mA**. Fast buses want the stronger pull-up; low-power designs want the weaker one.

DEEP DIVE · INTERNAL PULL-UPS ON A MICROCONTROLLER

Most microcontroller pins have a weak pull-up, and often a pull-down, built in that firmware can switch on, typically tens of kilohms. It saves a part on an undemanding line like a button input. For a bus with real speed or noise requirements, an external resistor of a value you chose is still the better call, because the internal one is weak and only loosely specified. So a button gets the internal pull-up, and an I2C bus gets external ones sized for the job.

THE SAME PIN THREE WAYS: PULLED UP TO HIGH, PULLED DOWN TO LOW, OR LEFT FLOATING.

On a One Thousand Drones board the I2C lines carry their pull-ups, and the microcontroller's boot-strap and enable pins carry pull-ups or pull-downs so the chip always starts in the state the design intends.

CHECKPOINT**1. What does a pull-up resistor do to an idle line?**

- a. Holds it at a known high level
- b. Speeds up the clock
- c. Blocks all current

ANSWER · A

A pull-up ties the line to the supply, so it rests high until something actively pulls it low.

2. An input pin connected to nothing does what?

- a. Reads a steady 0
- b. Floats and picks up noise
- c. Damages the chip

ANSWER · B

With nothing driving it, the pin has no defined level and stray coupling swings it around.

3. Which bus relies on pull-ups because its lines are open-drain?

- a. SPI
- b. UART
- c. I2C

ANSWER · C

I2C devices only pull low, so a pull-up on each line supplies the idle-high level.

- Prerequisite: resistors
- Related: I2C, the two-wire bus
- Next: digital isolation