

Teensy USB Switch Wiring

Adafruit TS3USB30 breakout, wired so the Teensy's single Micro-USB port can be handed off between the Raspberry Pi and a programming cable with one toggle switch.

The idea

Only two things actually pass through the manual DPDT toggle:

- **VBUS** (pole 1) — so the Pi's 5V and the programming PC's 5V can never be tied together.
- **The chip's S-select pin** (pole 2) — mechanically ganged to the same toggle lever, so S can never disagree with which VBUS is actually live.

D+/D- for both hosts land straight on the chip's own pads and are switched *inside* the TS3USB30 itself; GND runs straight through to everything, common and never switched.

Board pinout (confirmed against the physical board, 2026-10-14)

The TS3USB30 breakout has three pin groups, exactly as silkscreened:

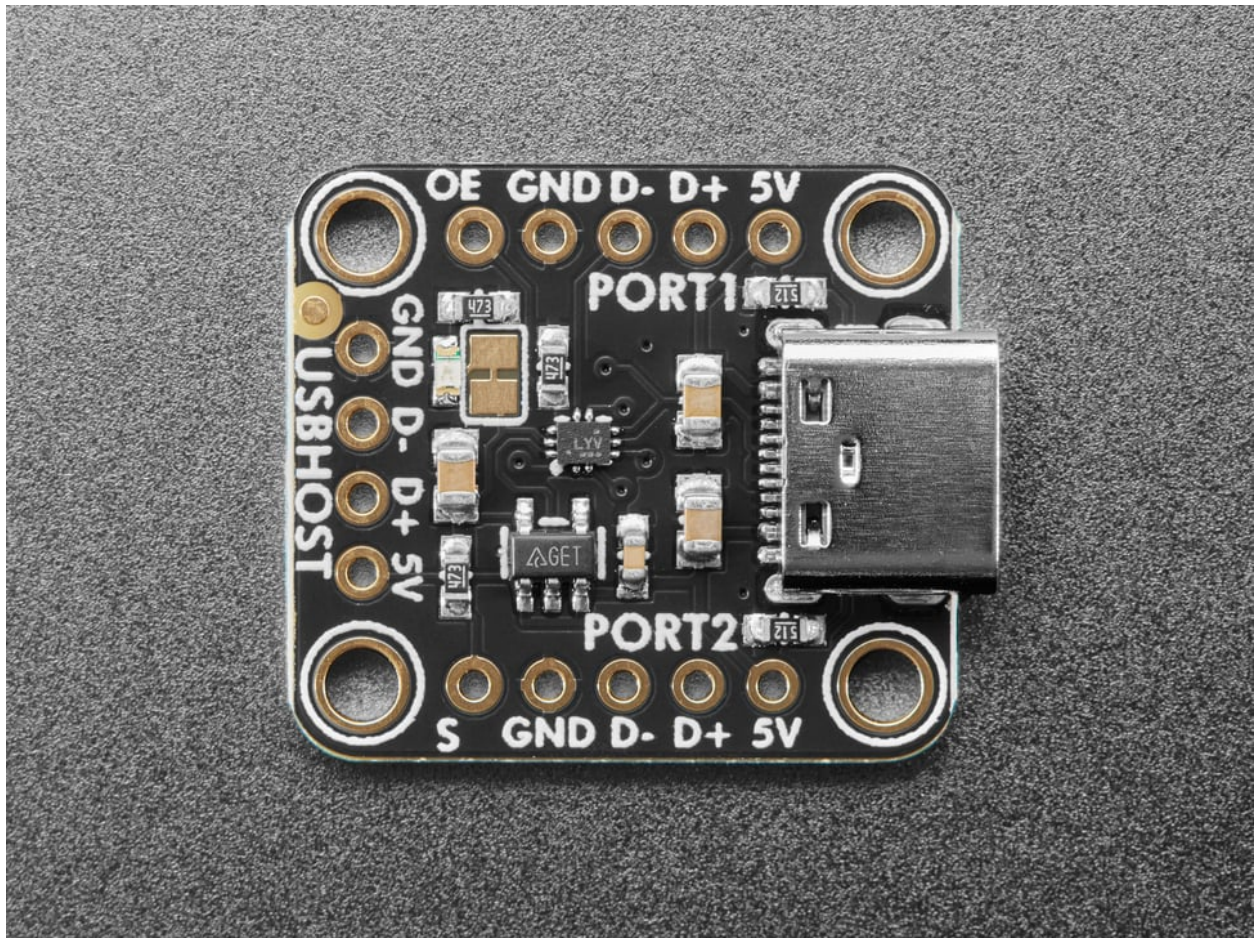


Figure 1: Adafruit TS3USB30 breakout board, pins labeled OE/GND/D-/D+/5V on the PORT1 row, GND/D-/D+/5V on the USBHOST row, and S/GND/D-/D+/5V on the PORT2 row

Group	Pins	Role here
PORT1 (top edge) USBHOST (left edge)	OE, GND, D−, D+, 5V GND, D−, D+, 5V	D1 — wired to the Pi Common/output side — also the board’s onboard USB-C jack; goes to the Teensy
PORT2 (bottom edge)	S, GND, D−, D+, 5V	D2 — wired to the programming PC

There is no separate “VCC” pad on this board — only the three **5V** pins above. Each group’s 5V pins (PORT1, USBHOST, PORT2) are tied together as one shared net on the board, and likewise each group’s GND pins are one shared net. That’s exactly *why* VBUS needs its own external switch pole (below) — the board itself does not switch power, only D+/D− and only inside the chip.

Note: OE lives on PORT1’s row and S lives on PORT2’s row — not generically “near the chip.” Keep that in mind when soldering: the OE ground-tie and the S-to-toggle wire land on opposite ends of the board.

Why VBUS gets its own switch pole, not the board’s built-in pass-through

Adafruit’s own board ties the common side’s power straight through to both downstream ports, because it’s designed for one host feeding two peripherals. Used in reverse — two hosts, one device — that would tie the Pi’s 5V rail directly to the programming computer’s 5V rail the moment both cables are plugged in, through that same shared 5V net described above. Routing VBUS through its own external switch pole keeps the two hosts’ power rails from ever touching; only the currently-selected host’s 5V reaches the board and the Teensy. The flip side of this: each host cable’s own VBUS wire goes **only** to the toggle’s throw terminals, never directly to PORT1 or PORT2’s own 5V pin — wiring it there would just recreate the short the switch exists to prevent, since those pins are the same net as everything else.

Parts

- Adafruit TS3USB30 1-to-2 USB switch breakout
- DPDT (double-pole double-throw) toggle switch — handles both the VBUS pole and the S-select pole together, so one flip always keeps them in agreement
- 2× Micro-USB cable (cut open, or a Micro-USB breakout pigtail) — one for the Pi, one for the programming computer
- 1× USB-C-to-Micro-USB cable — from the breakout’s own onboard USB-C jack to the Teensy

Wiring

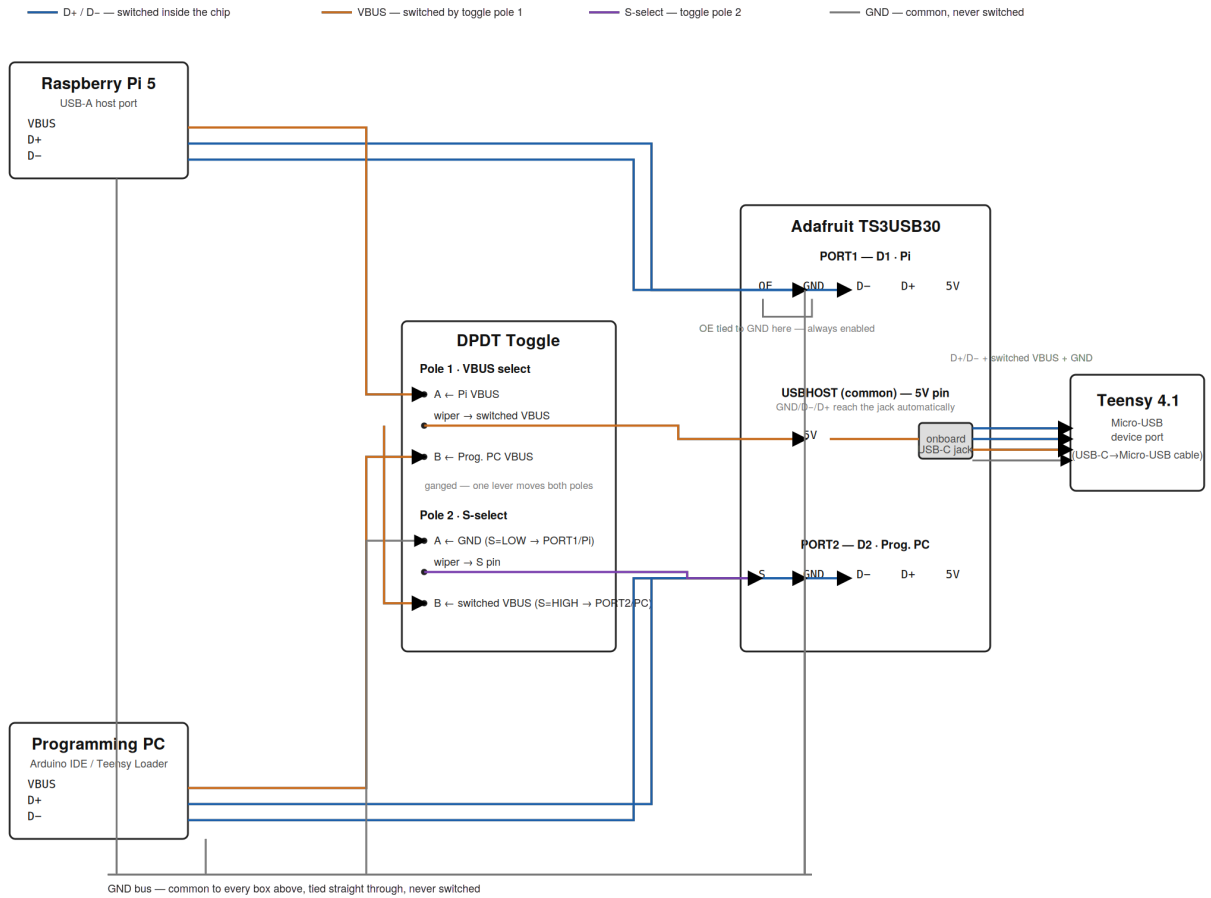


Figure 2: Wiring diagram: Raspberry Pi 5 and Programming PC each send D+/D- straight to the TS3USB30's PORT1 and PORT2 respectively, while their VBUS wires go only to the DPDT toggle, whose switched output feeds the USBHOST 5V pin and, through the board's onboard USB-C jack, the Teensy 4.1. A second toggle pole drives the S-select pin so it always agrees with which VBUS is live. GND runs as one common bus to every box.

From	Wire	To	Notes
Pi's Micro-USB cable	D+ / D-	PORT1 D+ / D-	Straight to the chip — not switched by the toggle
Pi's Micro-USB cable	VBUS	Toggle pole 1, throw A	—
Programming cable	D+ / D-	PORT2 D+ / D-	Straight to the chip — not switched by the toggle
Programming cable	VBUS	Toggle pole 1, throw B	—

From	Wire	To	Notes
Toggle pole 1, wiper (switched VBUS)	—	USBHOST 5V pin	Feeds the common side and, via the board's onboard USB-C jack, the Teensy. Leave PORT1's and PORT2's own 5V pins unconnected — same net, no need to double-wire them.
Shared GND bus	GND	Any one GND pin (e.g. PORT1's)	One common ground, never switched — all three groups' GND pins are already tied together on the board
PORT1's own GND pin	—	PORT1's OE pin	Active-low — tying OE to GND leaves the switch permanently enabled
Toggle pole 2, throw A	GND	—	S = LOW → selects D1 (Pi)
Toggle pole 2, throw B	Switched VBUS (same node as pole 1's wiper)	—	S = HIGH → selects D2 (programming PC)
Toggle pole 2, wiper	—	PORT2's S pin	Selects which side's D+/D− reaches the common port
USBHOST's onboard USB-C jack	D+/D− + switched VBUS + GND	Teensy's Micro-USB port	Via an off-the-shelf USB-C-to-Micro-USB cable — no need to wire USBHOST's own header pins separately, the jack carries the same net

Both throws of pole 2 are fed from signals pole 1 already switched, or from ground — no separate logic pin or pull-up is needed, and the two poles are physically locked together on one toggle lever, so S can never disagree with which VBUS is live.

Flipping the toggle: throw A puts S low and connects the Pi's VBUS — the Teensy enumerates against the Pi. Throw B puts S high and connects the programming computer's VBUS — the Teensy enumerates against Arduino IDE/Teensy Loader instead. Either way the Teensy sees a clean power-up, the same as if you'd unplugged one cable and plugged in the other by hand — there's no hub chip anywhere in the path.