LOGIC BOOLE 16 QUICK START

Revision C · from the box to a first capture

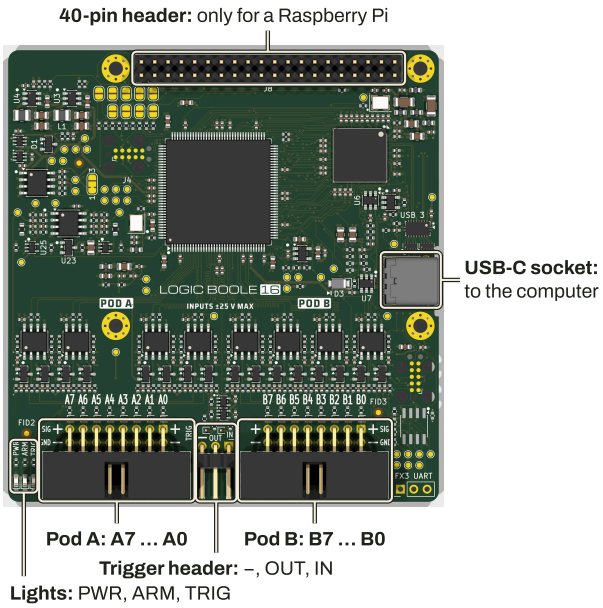
WHAT YOU NEED

In the box

- [x] **The board.** Nothing else: no cable, no leads, no spacers.

You bring

- [ ] **A computer:** a Mac, or Linux (Debian, Ubuntu, Raspberry Pi OS). Windows is not supported.
- [ ] **A USB-C cable made for USB 3** (5 Gbit/s; often marked SS or 5 Gbps), to a USB 3 port of the computer: USB-C to USB-C, or USB-C to a blue USB-A socket. This one cable carries power and data. A USB 2 cable or port also works, for slow signals: the board then carries about a ninth as much.
- [ ] **Leads for your signals:** jumper wires with sockets for 2.54 mm pins, or a 16-way (2 × 8) ribbon cable with an IDC socket for each pod.
- [ ] **Python 3.9 or later** and **libusb** on the computer (step 1). There is nothing else to install, and no driver.
- [ ] **For a Raspberry Pi only:** four M2.5 × 16 mm spacers and eight M2.5 × 6 mm screws.



1 GET THE COMPUTER READY (ONCE)

Mac. Install libusb with Homebrew:

```
brew install libusb
```

`python3 --version` should say 3.9 or later. If it is missing, `brew install python`.

Linux, Raspberry Pi OS. Python and libusb are already there. After the download below, allow your user to open the board, once:

```
sudo cp software/udev/60-logic-boole.rules
/etc/udev/rules.d/
sudo udevadm control --reload
sudo udevadm trigger
```

Then download the Logic Boole 16 repository from **github.com/mgwalkerjr95/logic-boole-16** (Download ZIP, or `git clone`), and open a terminal in its folder. Nothing in it needs building or installing.

2 PLUG THE BOARD IN

Connect the board's USB-C socket to a USB 3 port of the computer. Leave the probes unconnected for now.

> **Mac laptops** ask "Allow accessory to connect?" for a USB device they have not seen: choose **Allow**. macOS can ask once more in step 3, when the board comes back to USB under its own name.

3 CHECK IT

```
./logicboole check
```

This loads the board and captures a test pattern that the board makes itself. In a few seconds it prints:

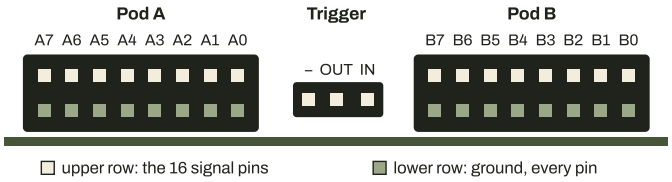
```
Python 3.12.4: fine.
libusb: found.
The board: on USB.
The firmware: downloaded, checked and kept on this computer.
Loaded: firmware version 2, FPGA design version 4, serial number 0009021E1311241C.
USB: 5 Gbit/s.
A capture of the FPGA's test pattern, 100000000 samples at 200 MS/s: every word right.
A capture of the FPGA's test pattern, 2000000 samples at 10 MS/s: every word right.
The board is ready.  "logicboole" opens the app.
```

If something is missing, its line says what to do. The board keeps no firmware: the software loads it at every plug-in, and downloads it the first time (the computer needs the internet that once). Once loaded, the ARM light blinks once a second.

4 CONNECT YOUR CIRCUIT

- **Ground first.** Join any pin of a pod's lower row to your circuit's ground. Every pin of the lower rows is ground.
- **Then the signals,** to pins of the upper rows: A0 to A7 on pod A, B0 to B7 on pod B. The names are printed on the board above the pods. Each input takes up to ±25 V.
- Each pod has its own threshold, so pod A can watch 3.3 V logic while pod B watches 1.8 V.
- For signals faster than a few megahertz, keep the leads short and put a ground lead beside the signal.
- The trigger header is 3.3 V logic: – is ground, OUT goes high when a capture triggers, IN can be part of a trigger.

Looking into the pods from the board's edge, parts side up



5 TAKE A CAPTURE IN THE APP

```
./logicboole
```

opens the app in your browser, at `http://localhost:8016`, with the board as its source.

- **Set the two thresholds** to about half your logic voltage: 2.5 V for 5 V logic, 1.65 V for 3.3 V, 0.9 V for 1.8 V, 0.6 V for 1.2 V. They go from 0.3 V to 3 V.
- **Pick a rate and a length.** Ten times your fastest signal or more is a good rate to start with; the fastest is 200 MS/s, a sample every 5 ns. Then **press Start**, or the space bar.
- **Auto-detect** looks for I²C, SPI and serial in the capture and decodes them. Point at a pulse for its width and frequency.
- To wait for an event, pick a channel and an edge under Trigger before you press Start.
- No circuit at hand? `./logicboole --fake` runs the app with a stand-in board that makes an I²C bus.

6 GO FURTHER

`./logicboole capture bus.sr --rate 10M --time 2 -a 1.65 -b 0.9` takes a capture from the command line into a file for PulseView. `./logicboole selftest` is the long check of the board: 34 captures, about half a minute. `software/README.md` in the repository has every command, the triggers, the protocol decoders, the Python package, and letting an AI agent use the board.

IF SOMETHING DOES NOT WORK

What you see	What to do
"The board: not on USB."	Try another cable and port. A cable made only for charging carries no data.
"macOS has not let it connect"	Run it again and choose Allow when macOS asks. Setting: "Allow accessories to connect".
"this user may not open it"	Linux: install the USB rule of step 1, then unplug the board and plug it in again.
"USB: 480 Mbit/s"	The cable or the port is USB 2. The board works; fast signals need a USB 3 cable and port.
A capture ends early: "overflow"	Too many changes for the link: lower the rate, shorten the capture, or use USB 3.
An input stays at 0 or at 1	Check the ground lead and that pod's threshold.
Anything else odd	Unplug the board and plug it in again: it is then as it was.

ON A RASPBERRY PI 4 OR 5

- Set the board on the Pi's 40-pin header, on four M2.5 × 16 mm spacers held by eight M2.5 × 6 mm screws (yours to get: they are not in the box); it takes its power there. Join its USB-C socket to a blue USB 3 socket of the Pi with a short USB 3 cable, for the data.
- Then steps 1 to 6 as on any Linux computer. With no screen on the Pi: `./logicboole --host 0.0.0.0`, and open `http://` the Pi's name `:8016` elsewhere, on a network you trust (the app has no password).

© 2026 Michael Walker · open hardware and software: see the repository's README for the licenses