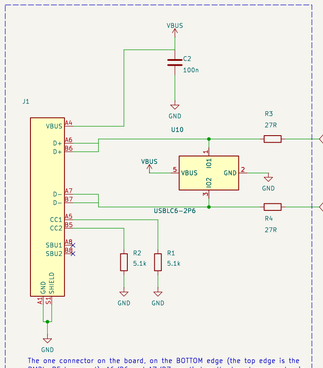
bluebuzz-rm2 — generated schematic

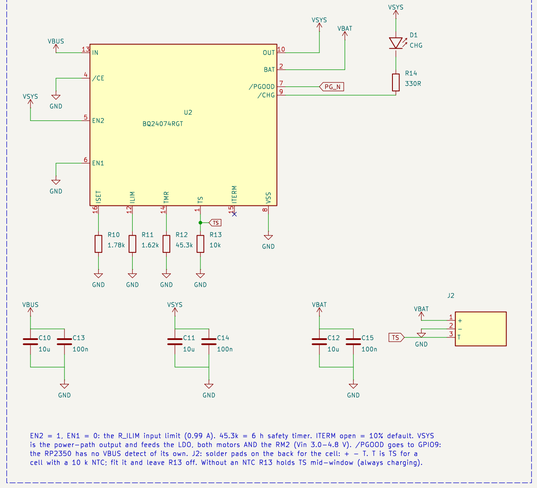
109 components / 72 nets / 6-layer. Bluetooth-audio vibrator controller: RP2350A + Raspberry Pi RM2 (CYW43439); USB-C, every other lead on back-side pads; BQ24074 charger with power path, a soft power latch (button, GPIO hold, USB), two low-side motor channels, 4 DPS368 barometers on the back (four corners and the centre, a tactile skin) on a switched 3.3 V rail, an IMU and a 1010 RGB status LED. Rails are power symbols; cross-block signals are global labels. Generated from scripts/circuit.py; rebuild with `python

1 — USB-C and ESD



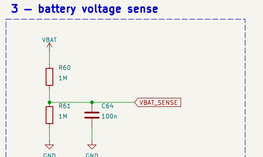
The one connector on the board, on the BOTTOM edge (the top edge is the RM2's RF keep-out). A6/B6 and A7/B7 are tied or the board enumerates in one cable orientation only; 5.1k on each CC. The USBLC6-2P6 clamps VBUS and protects D+/D−; 27 R in each line at the RP2350 is the reference design's.

2 — BQ24074 charger with power path



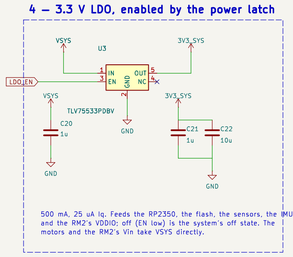
EN2 = 1, EN1 = 0: the R_ILIM input limit (0.99 A). 45.3k = 6 h safety timer. ITERM open = 10% default. VSYS is the power-path output and feeds the LDO, both motors AND the RM2 (Vin 3.0–4.8 V). /PGOOD goes to GPIO9; the RP2350 has no VBUS detect of its own. J2: solder pads on the back for the cell: + — T. T is TS for a cell with a 10 k NTC; fit it and leave R13 off. Without an NTC R13 holds TS mid-window (always charging).

3 — battery voltage sense



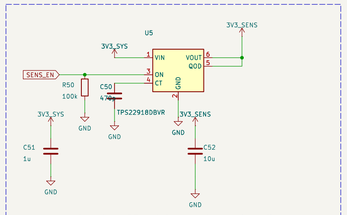
1M/1M, 2.1 uA, into ADC0 (GPIO26). C64 feeds the ADC's sampling charge; the RP2350's ADC input is 500 ohm-ish during a sample, so oversample.

4 — 3.3 V LDO, enabled by the power latch



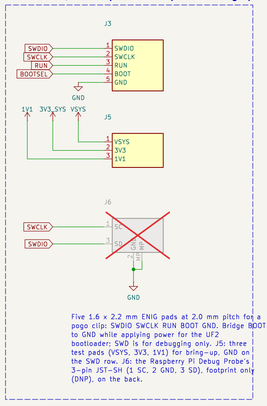
500 mA, 25 uA Iq. Feeds the RP2350, the flash, the sensors, the IMU and the RM2's VDDIO; off (EN low) is the system's off state. The motors and the RM2's Vin take VSYS directly.

5 — switched sensor rail



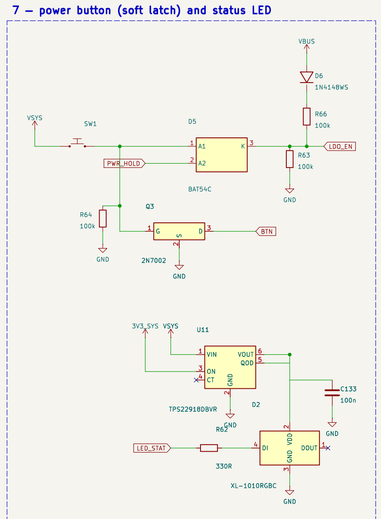
The switched 3.3 V rail for the tactile array (block 6): the sensors draw little, but gating them lets firmware power-cycle the array and keeps the rail off in sleep.

6 — SWD / boot pads, test pads, debug-probe connector (back)



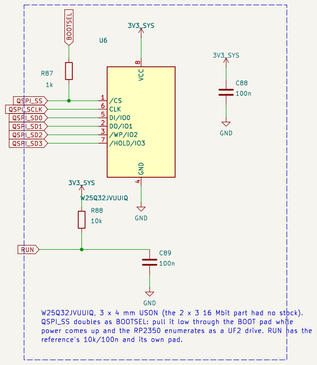
Five 1.6 x 2.2 mm ENIG pads at 2.0 mm pitch for a pogo clip: SWDIO SWCLK RUN BOOT GND. Bridge BOOT to GND while applying power for the UF2 bootloader; SWD is for debugging only. J5: three test pads (VSYS, 3V3, 1V1) for bring-up. GND on the SWD row. J6: the Raspberry Pi Debug Probe's 3-pin JST-SH (1 SC, 2 GND, 3 SD), footprint only (DNP), on the back.

7 — power button (soft latch) and status LED



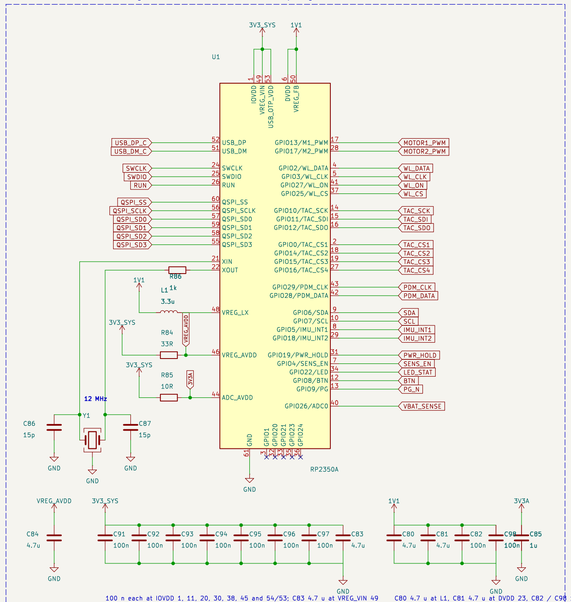
Press SW1: VSYS reaches the LDO's EN through D5 and 3V3 comes up; the RP2350 boots and drives PWR_HOLD (GPIO19) high, so the board stays on after release. Firmware reads the button on GPIO8 (Q3, active low, internal pull-up) and switches off by driving PWR_HOLD low. R63 holds EN off; R64 discharges the gate. USB in (VBUS through D6 and R66) also holds EN on, so flashing and SWD need no button. Charging runs while off. Status: a 1010 addressable RGB LED (WS2812B type) on GPIO22 through 330 R; its VSYS supply goes through U11 (a second TPS22918, ON = 3V3_SYS), so it draws nothing while the system is off.

8 — QSPI flash, reset and boot



W25Q32JVUUIQ, 3 x 4 mm USON (the 2 x 3 16 Mbit part had no stock). QSPI_SS doubles as BOOTSEL: pull it low through the BOOT pad while power comes up and the RP2350 enumerates as a UF2 drive. RUN has the reference's 10k/100n and its own pad.

9 — RP2350A: core regulator, clock, USB, decoupling

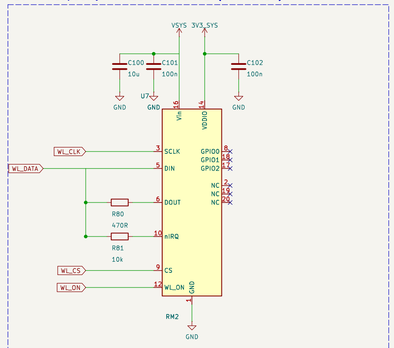


Raspberry Pi's RP2350A minimal design: 3.3 uH on VREG_LX with 4.7 u at the output and at DVDD, 100 n at the third DVDD, 4.7 u on VREG_VIN, VREG_AVDD through 33 R + 4.7 u, ADC_AVDD through 10 R + 1 u, 27 R in each USB line, 12 MHz crystal with 1 k in series with XOUT, the reference's ABM8-272-T3 (CL 10 pF) with its 15 pF loads. The radio's gSPI is on GPIO2/3/23/25 (the board file overrides the SDK's data and clock pins).

100 n each at IOVDD 1, 11, 20, 30, 38, 45 and 54/53; C83 4.7 u at VREG_VIN 49

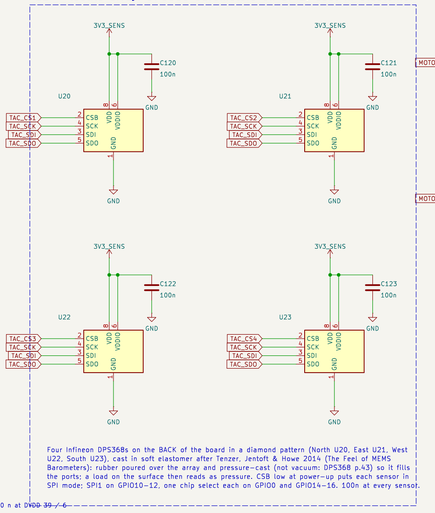
C80 4.7 u at L1; C81 4.7 u at DVDD 23. C82 / C98 100 n at DVDD 39 / 6

10 — Raspberry Pi Radio Module 2 (CYW43439)



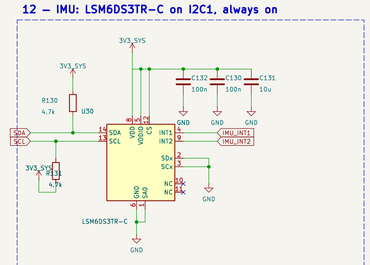
RMC20452T: Wi-Fi 4 + Bluetooth 5.2 Classic/LE, PCB antenna, modular certification. Vin 3.0–4.8 V from VSYS (271 mA TX peaks stay off the LDO), VDDIO 3V3, gSPI: GPIO3 clock, GPIO2 data (DIN direct, DOUT via 470 R, nIRQ via 10 k), GPIO25 CS, GPIO27 WL_ON on pads 12 and 13 (WL_REG_ON and BT_REG_ON). The datasheet's RF keep-out — 5 mm into the body, 10.5 mm either side, 13.5 mm beyond — is a rule area on every layer; the module's antenna end is the board edge.

11 — tactile array: DPS368 barometers on SPI1, cast in elastomer



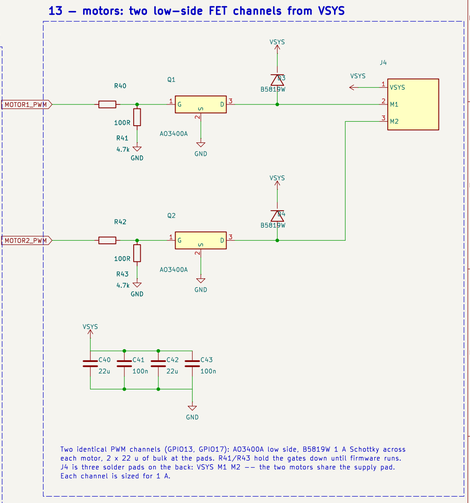
Four Infineon DPS368s on the BACK of the board in a diamond pattern (North U20, East U21, West U22, South U23), cast in soft elastomer after Tenzer, Jentoft & Howe 2014 (The Feel of MEMS Barometers): rubber poured over the array and pressure-cast (not vacuum; DPS368 p.A3) so it fills the ports; a load on the surface then reads as pressure. CSB low at power-up puts each sensor in SPI mode; SPI1 on GPIO10–12, one chip select each on GPIO0 and GPIO14–16. 100n at every sensor.

12 — IMU: LSM6DS3TR-C on I2C1, always on



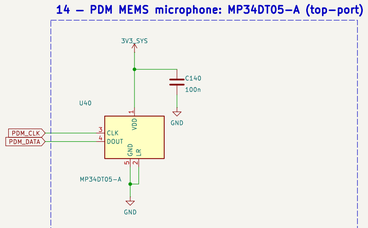
ST LSM6DS3TR-C six-axis IMU on I2C1 (GPIO6/7, 4.7k pull-ups to the always-on rail); CS high selects I2C, SA0 low gives address 0x6A; the aux I2C (SDx/SCx) is unused and tied to GND. INT1/INT2 to GPIO5/18 for wake-on-motion, tap and tilt. 100n + 10u on VDD, 100n on VDDIO.

13 — motors: two low-side FET channels from VSYS



Two identical PWM channels (GPIO13, GPIO17): AO3400A low side, B5819W 1 A Schottky across each motor. 2 x 22 u of bulk at the pads. R41/R43 hold the gates down until firmware runs. J4 is three solder pads on the back: VSYS M1 M2 — — the two motors share the supply pad. Each channel is sized for 1 A.

14 — PDM MEMS microphone: MP34DT05-A (top-port)



ST MP34DT05TR-A top-port digital PDM MEMS microphone (3 x 4 x 1 mm HCLGA-4, 64 dB SNR, 122.5 dBSPL AOP) on the always-on 3.3 V rail. LR tied to GND selects the left-channel slot; CLK on GPIO29, DOUT on GPIO28 (PIO PDM state machine). Top acoustic port needs no PCB hole.

Power flags — the ERC root for each rail



No copper, no netlist entry: every rail is fed through a pin the symbols declare passive, so ERC needs a driver per rail.

generated from scripts/circuit.py by gen_schematic.py
RP2350A + RM2 vibrator controller, DPS368 tactile array, LSM6DS3TR-C IMU
sequoia-hope
Sheet: /
File: bluebuzz-rm2.kicad_sch

Title: bluebuzz-rm2

Size: User	Date: 2026-10-18	Rev: A
KiCad E.D.A. 9.0.8		Id: 1/1