



JITX · JUMPSTART KIT #1 · PART 2

Four component families, no parts database

Yageo RC_L · Panasonic ERJ · Vishay CRCW · Samsung CL — straight from their datasheets

Drive jitx-component-modeler to build parameterized families where the class is the data — then ask for any part in the catalog and get an orderable MPN back.



Model the family, not the part

Part 2 teaches: datasheet tables become code. One class stands in for a manufacturer's whole catalog family — sizes, tolerances, power ratings, packaging — transcribed once from the datasheet.

No parts database, no network. Ask for 49.9 k Ω $\pm$ 1 % in 0402 and the class derives the dimensions, land pattern, ratings and the exact orderable MPN — entirely offline.

Fail fast, name the options. An invalid combination raises a ValueError that lists what the family actually offers — the datasheet's constraints, enforced at construction.

Abstraction earns its keep. Family 1 is built self-contained; shared helpers are extracted only when Family 2 proves the pattern repeats. Value encoders never merge — three vendors, three schemes.

A shared core, a thin layer per vendor

SHARED CORE

`chip_smt.py`

- ChipDims — Toleranced package dimensions
- chip-size land-pattern generator
- two-pin insert (series drop-in)
- E-series value check
- carry-correct significant rounding

Yageo RC_L

1

RKM code → 100K

MPN RC0402JR-07100KL

Panasonic ERJ

2

EIA 3-digit → 103

MPN ERJ3GEYJ102V

Vishay CRCW

3

fixed 4-char RKM → 10K0

adds a TCR axis

Samsung CL (MLCC)

4

3-char pF code → 104

adds dielectric + voltage axes

Shared: geometry, insertion, validation helpers. Never shared: the vendor's value encoding and part-number grammar.

What you build

Family 1 · Yageo RC_L, self-contained — eleven chip sizes, tolerance and packaging axes, RKM value codes. No helpers yet — nothing has repeated.

Family 2 · Panasonic ERJ, extract the core — the second family proves what's shared; chip_smt.py is born and Family 1 is refactored onto it.

Family 3 · Vishay CRCW — adds the TCR axis, and takes the generator's standard EIA dimensions — asserted per size against the datasheet's own table.

Family 4 · Samsung CL, the capacitor — the pattern generalizes across component type — dielectric and voltage axes, capacitor symbol, same shared core.

Proof by part number — RC0402JR-07100KL · ERJ3GEYJ102V · CRCW0603562RFKEA · CL10B104KB8NNNC — each encodes to the vendor's own catalog grammar.

Verification is the key

Datasheet spot-checks. Dimensions, ratings and codes are hand-read from the four datasheets and asserted in tests — per family, before it counts as done.

MPN cross-checks. Generated part numbers match the vendor's catalog listings character for character.

E-series discipline. Requested values validate against the tolerance's E-series grid, with carry-correct rounding before encoding.

Gates at every step. pyright → pytest → jitx build for every family, and jitx-code-review before the final commit.

The combined design. All four families instantiate together in one design that builds successfully — the showcase the closing prompt answers from.

Run it

The runbook: `jumpstart-kits/js1-stackup-components/part2-parametric-passives/` — paste `JS1_Part2_Parametric-Passives_Runbook.md` into your agent and tell it to follow the file. Two [HUMAN] gates stop for you.

The ground truth: four manufacturer datasheets, fetched by URL during the run into a gitignored working folder — linked, never committed.

The reference solution: `jitxexamples.jumpstart_kits.js1_stackup_components.parametric_passives` — four families, the shared core, tests, and buildable designs.



CLOSE THE LOOP

“give me a 49.9 k Ω 0402 1 % Yageo and a 100 nF X7R 0603 50 V Samsung CL”

Two real, orderable parts — MPNs included — answered from code.