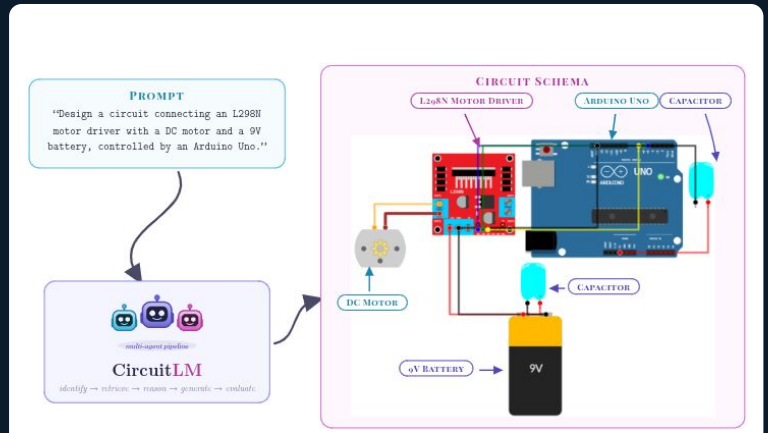
CircuitLM

A Multi-Agent LLM-Aided Design Framework for Generating Circuit Schematics from Natural Language Prompts

Khandakar Shakib Al Hasan · Syed Rifat Raiyan · Hasin Mahtab Alvee
Wahid Sadik

Dept. of Computer Science & Engineering · Dept. of Electrical & Electronic Engineering
Islamic University of Technology · Dhaka, Bangladesh



Natural-language prompt ⟶ structured, renderable schematic

MOTIVATION

LLMs can describe a circuit. Can they design one?

Ask a frontier model to “build a gesture-control circuit with an ESP32-C3 Mini and five MPU6050 sensors,” and you get prose — ambiguous, incomplete, and electrically unsound.

01

Hallucinated parts

Component names and pin labels that do not exist on any real device.

02

Violated physics

Missing pull-ups, decoupling caps, current-limiting resistors; ignored power-distribution constraints.

03

Unusable output

Free-form prose, not machine-readable — incompatible with Fritzing or Proteus.

Consequence: the engineer must rebuild the design by hand in an EDA tool — a slow step that surfaces yet more faults. LLM-generated circuits are rarely functional or safe on the first pass.

RESEARCH GAP

Existing benchmarks measure the wrong thing

PINS100 and MICRO25 score textual plausibility. None of them verify that the circuit is physically buildable.

1. **Holistic electrical validity**
   No check for shorts, logic-level mismatch, or floating inputs.

2. **Structured machine-readable output**
   Nothing a downstream tool can consume.

3. **Automated visualisation**
   No way to see — or correct — what was generated.

SO WHAT WE NEED IS

Grounding + verification

Retrieval that pins generation to real components, and a checker that decides — deterministically — whether the result is safe to build.

That is CircuitLM.

CONTRIBUTIONS

Four contributions

A CircuitLM pipeline

A five-stage multi-agent system that converts natural language into executable schematics with interactive visualisation.

B CircuitJSON

An open, tool-agnostic schematic interchange format separating structural connectivity from parametric configuration.

C Deterministic ERC benchmark

A graph-based Electrical Rule Checking engine plus a 100-prompt, difficulty-stratified benchmark dataset.

D Retrieval that scales

Decoupling prompt context from library size N gives O(k) prompt complexity for k required components.

01

The Pipeline

Five stages, each targeting one distinct failure mode

ARCHITECTURE

System overview

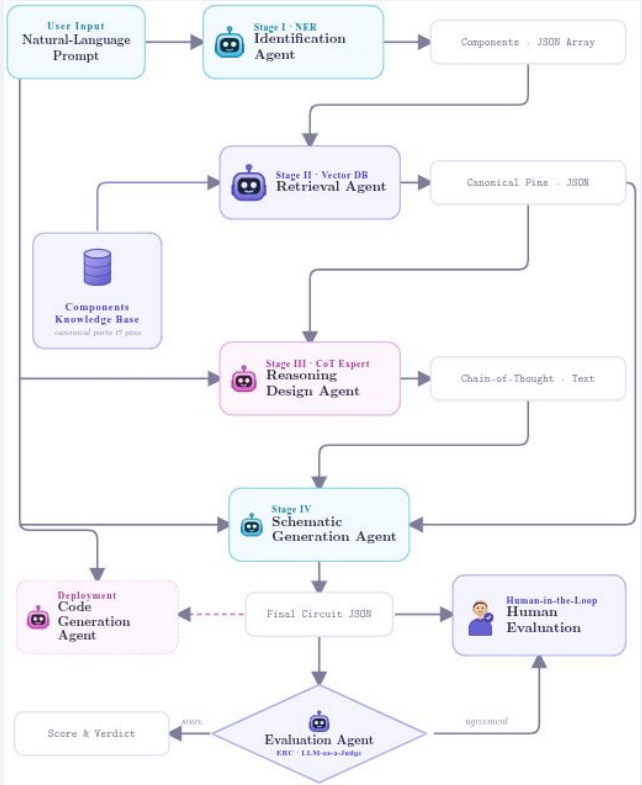


Figure 2 — the CircuitLM framework, end to end.

I — Identification Agent
NER + intent → generic component categories

II — Retrieval Agent
Vector search over a canonical component KB

III — Reasoning Agent
Chain-of-Thought electrical design document

IV — Generation Agent
Strictly formatted CircuitJSON schema

V — Schematic Visualizer
Interactive SVG, human-in-the-loop

Plus: an Evaluation Agent (ERC + LLM-as-a-Judge) and an optional firmware code-generation agent.

STAGES I & II

Grounding: stop the model from inventing parts

I Identification Agent

NER plus intent analysis over the prompt. Deliberately limited to generic categories — not full part records — to shrink the search space and the token budget.

["Microcontroller", "Motor Driver", "DC motor"]

II Retrieval Agent

Embedding search over a ChromaDB knowledge base of 70+ canonical components, merged with a fuzzy alias lookup. Returns verified names and exact pin-outs — so pins are read from the database, never generated.

SAFETY GUARDRAIL

Out-of-Distribution halt

If a requested part is absent from the knowledge base, an OOD flag is raised, execution stops, and a human is asked to review. We would rather refuse than wire something hallucinated.

~2%

of prompts triggered an OOD halt — the guardrail fires rarely on in-scope designs.

O(k)

prompt complexity in the k parts a design needs — independent of library size N.

STAGES III & IV

Reason first, then serialise

Separating engineering reasoning from JSON synthesis preserves electrical logic under strict grammar constraints.

III Reasoning Design Agent

The cognitive core. Given the prompt, the validated part list, and canonical pins, it emits a structured Chain-of-Thought document.

- Functional goals
- Power requirements and voltage domains
- Safety constraints
- Pin-level wiring logic

Least-to-most prompting decomposes complex requirements into hierarchical sub-tasks before any code is written.

IV Circuit Generation Agent

Transforms the CoT document into CircuitJSON — a strictly formatted, machine-readable schematic object inspired by Wokwi.

CIRCUITJSON CARRIES

Exact component placement — coordinates and rotation
A canonicalised list of pin-to-pin connections
An attrs field for electrical parameters and limits

Structure and parameters stay separate: wiring is deterministic, component values remain configurable.

STAGE V

Visual-first, human-in-the-loop

CircuitJSON renders into interactive SVG. Known components map to predefined symbols; unknown items become labelled glyphs.

Force-directed placement

Groups connected parts and prevents overlap.

Manhattan wire routing

Short paths, minimal bends.

Direct manipulation

Users drag components and reshape wires in real time.

Because layout is interactive, optimising edge crossings is out of scope — we evaluate electrical reasoning, not graph drawing.

Raw autonomous output — no manual adjustment

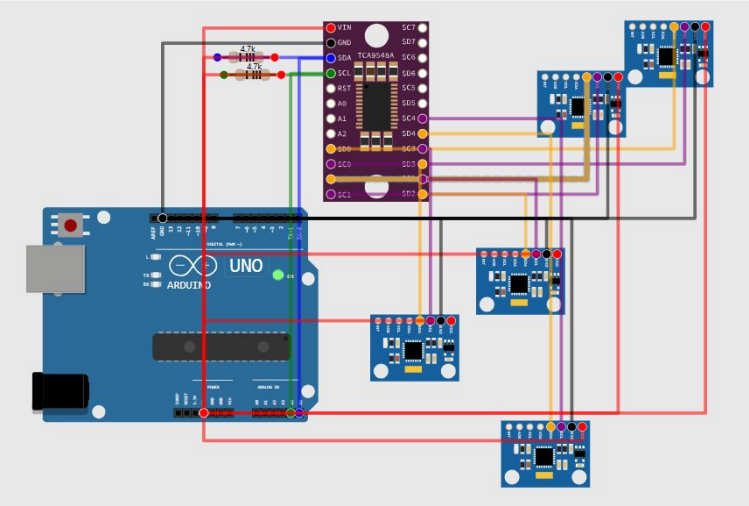


Figure 4 — sign-language glove: five IMU sensors, one per fingertip, rendered by the visualisation engine.

02

Evaluation

100 prompts · 5 frontier LLMs · two verification layers · blind expert audit

METHODOLOGY

Two verification layers, one strict metric

1 Deterministic ERC engine

The schematic becomes a bipartite graph of components, pins and galvanic nets in networkx. Weighted traversal distinguishes a direct short from a resistive bridge.

Checks: VCC–GND shorts · current-limiting and pull-up resistors · flyback diodes · logic-level compatibility (Vout > Vin,max) · floating inputs · PWM pin capability

2 LLM-as-a-Judge meta-evaluator

ERC proves connectivity, not intent. A separate QA agent — Claude Sonnet 4.5, excluded from the generator pool to avoid self-preference bias — audits protocol correctness and logical routing.

It also assumes board-level reality: breakout modules often carry their own regulators and support circuitry.

PRIMARY METRIC

Pass@1

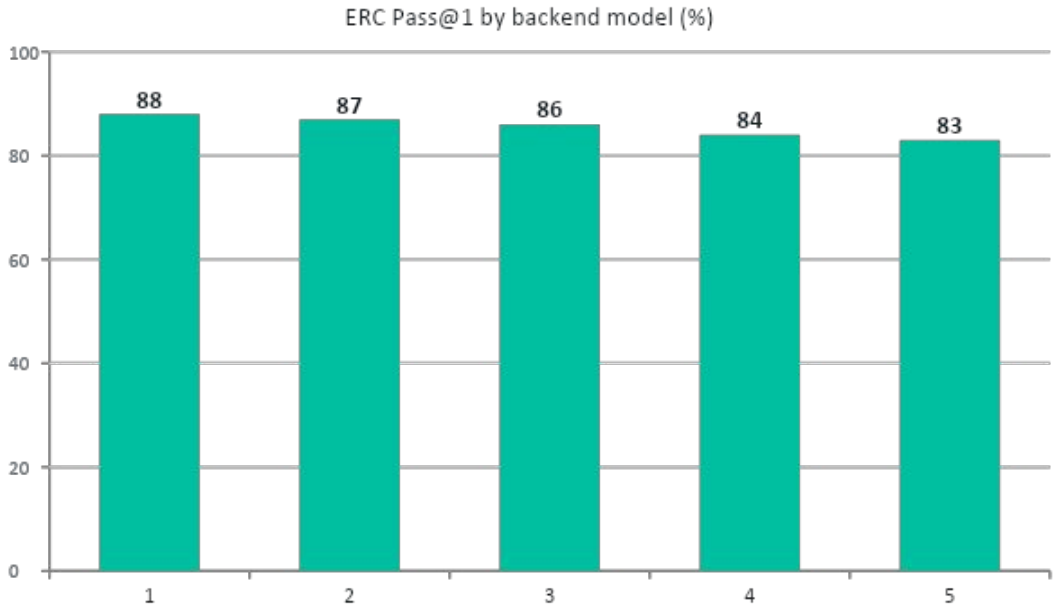
A pass requires zero Fatal and zero Major errors. Minor errors and warnings are logged, not penalised. OOD halts count as failures.

SEVERITY TIERS

Fatal	Immediate damage — e.g. VCC–GND short
Major	Guaranteed failure — logic mismatch, no flyback diode
Minor	Valid but poorly chosen passive values
Warning	Convention deviations — missing decoupling caps

Blind expert audit — three calibrated EE experts: Fleiss' $\kappa = 0.82$ (Fatal 0.94, Major 0.88)

Deterministic ERC: near-elimination of fatal faults



Mean of three runs per model, temperature 0.

FATAL ERRORS PER PROMPT

μ = 0.0

for nearly all backends. The CoT planning phase reliably forces models to identify VCC/GND rails and avoid critical shorts.

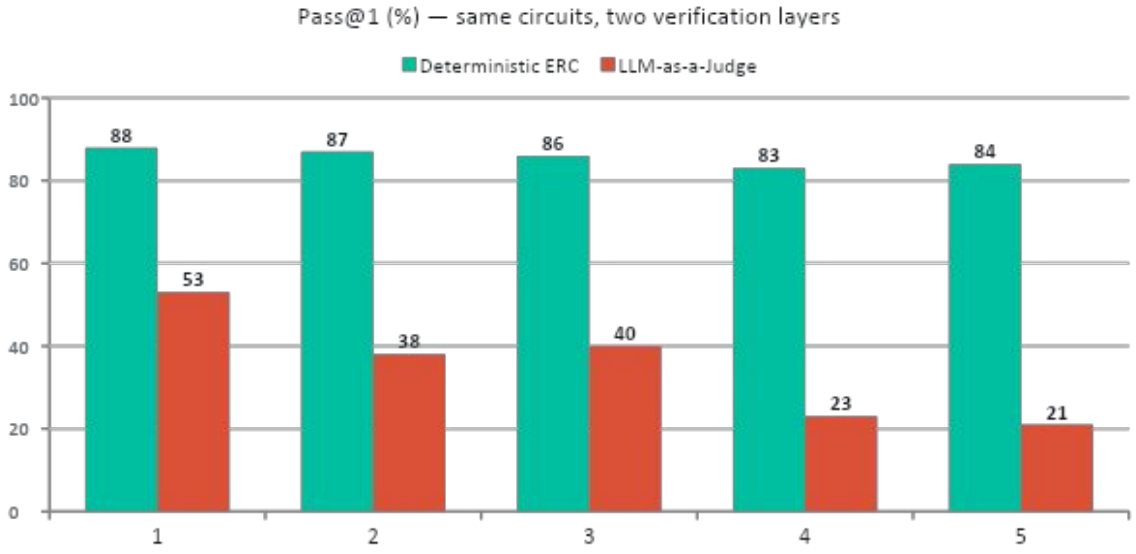
Dominant remaining failure

Major errors — logic-level mismatch and missing current-limiting resistors in multi-component systems. Models overlook voltage-domain boundaries when 5 V and 3.3 V parts are mixed.

RESULTS · LAYER 2

The evaluation gap

Passing electrical rule checks is not the same as being a working design. The two layers disagree — sharply.



GEMINI 2.5 FLASH

88% → 53%

Llama falls further still: 84% → 21%.

Is the judge hallucinating faults?

No. A blind expert audit of the judge's own outputs gives Fleiss' κ = 0.78 (95% CI 0.65–0.88) — it is reliably finding real, protocol-level faults.

ERROR ANALYSIS

What rule-based checking misses

Protocol direction

UART TX wired to TX; SPI MOSI/MISO swapped or bus pull-ups omitted.

Pin capability

PWM assigned to a non-PWM pin; multiplexed pins already reserved by an active SPI or UART bus.

Address collisions

Two devices sharing one I²C address on the same bus.

Brownout risk

A 3.3 V GPIO driving a 5 V inductive load — inside logic tolerance, functionally invalid.

Case study — MOSFET solenoid driver

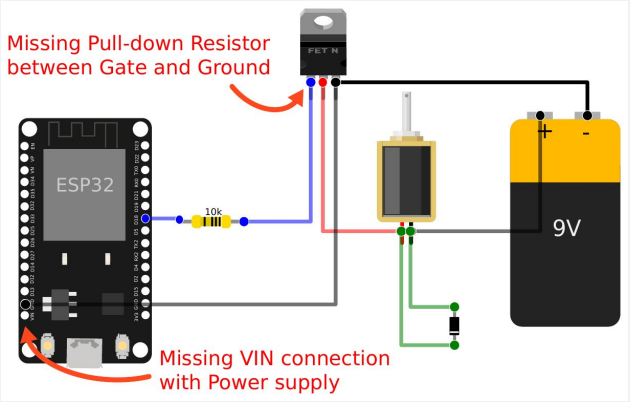


Figure 5 — the schema is structurally compliant, yet the QA agent flags a missing gate pull-down and an unconnected VIN.

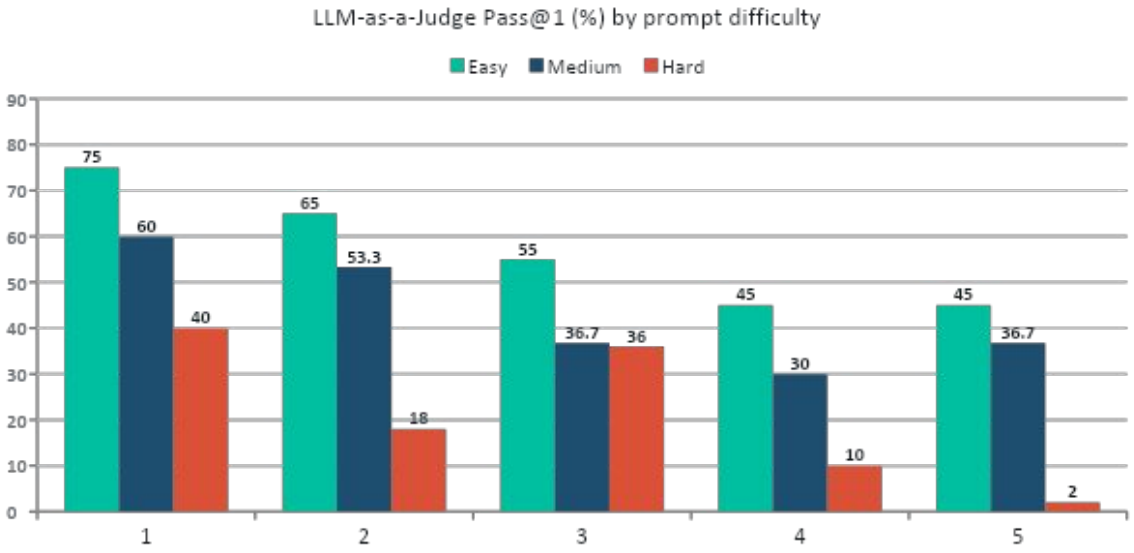
Prompt: "How do I wire an N-channel MOSFET to drive a solenoid using PWM signal?"

Every one of these passes continuity and logic-level checks. None of them works on a bench.

RESULTS · STRATIFIED

Difficulty is where the gap lives

The 100-prompt benchmark is tiered by component count, distinct protocols, and number of voltage domains.



HARD TIER

40% → 2%

Spread from best to worst backend. Under ERC the same tier looks uniformly healthy at 84–92%.

Reading this

Complexity is not spread evenly across models. Backend choice barely matters on easy prompts and decides the outcome on hard ones.

ABLATION

Is the pipeline earning its keep?

Strongest possible baseline: bypass retrieval and CoT entirely, and inject the full unfiltered component library — every description and pin definition — into one system prompt.

~15,000

tokens per request for the zero-shot full-library baseline for 70 components

~3,000

tokens per request with targeted retrieval and structured planning

WHY THE BASELINE CANNOT SCALE

Injecting an industrial library into a single prompt is cost-prohibitive and breaches context limits. Llama-3.3 could not produce zero-shot output at all — immediate context overflow.

ERC Pass@1 — baseline → CircuitLM

Deepseek v3.1	77% → 86%	(+9)
Qwen-3 235B	79% → 87%	(+8)
Gemini 2.5 Flash	84% → 88%	(+4)
GPT-5 mini	85% → 83%	(-2)

Where the CoT phase really pays

For frontier models the average gap is modest — they already reason internally. But remove the CoT agent and Gemini's hard-tier Pass@1 halves, 40% → 20%. GPT-5 mini improves slightly: for distilled models, prescriptive planning may interfere with internal heuristics.

LIMITATIONS

What we have not shown

Stated plainly, because each one is a concrete next step.

No simulator in the loop

We verify topology, not behaviour. Wokwi or Proteus co-simulation remains future work.

Single LLM evaluator

One judge model, so self-consistency and judge bias are not fully controlled.

Digital and embedded only

Analog circuit design is out of scope; SPICE-class analysis is not attempted.

No netlist export yet

Converting CircuitJSON to KiCad-compatible netlists needs further engineering.

Multi-agent overhead

Iterative queries and variable API latency cost roughly 8 s per design.

Curated 70-part library

Broad protocol coverage, but expanding to industrial scale is still a data-entry effort.

CONCLUSION

Grounding makes circuits buildable.
Semantic verification tells you whether they work.

1

Retrieval eliminates the failure mode ERC cares most about

Fatal errors reach $\mu \approx 0.0$ across five frontier backends, at one-fifth the token cost of full-library prompting.

2

Rule-based checking alone overstates quality

88% → 53% on identical circuits. Any future EDA benchmark needs a semantic layer, expert-validated

3

CircuitJSON as a shared bridge

We propose it as a standard interchange format between natural language and hardware description, and for future dataset creation.

Thank You