JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

COMPONENT SELECTOR & CIRCUIT SIMULATOR

[1]Khan Areeba Tehreem.[2] Prof.S.A.Khusro , [3]Prerana Vahule, [4]Patil Ayushi
[1]Student of Department of Computer Science and Engineering, P.E.S College of Engineering, Chh. Sambhajinagar.
[2]Professor Department of Computer Science and Engineering, P.E.S College of Engineering, Chh. Sambhajinagar.
[3]Student of Department of Computer Science and Engineering, P.E.S College of Engineering, Chh. Sambhajinagar.
[4]Student of Department of Computer Science and Engineering, P.E.S College of Engineering, Chh. Sambhajinagar.

Abstract: Selecting electronic components for a circuit requires engineers and learners to interpret datasheets, compare electrical limits, identify functionally suitable alternatives, and assess the effect of a proposed replacement on circuit behaviour. These tasks are often distributed across manufacturer documents, component catalogues, spreadsheets, and separate simulation tools. This paper presents the design of a web-based prototype named Component Selector and Circuit Simulator that brings component discovery, specification comparison, alternative-component assessment, and circuit-analysis workflows into one environment. The component-selection module organises parts using identity, category, manufacturer, part number, functional role, descriptive information, and electrical attributes such as operating voltage, input/output voltage, current rating, power rating, logic level, and frequency. A comparison workflow is intended to expose differences between a selected component and candidate alternatives, while a circuit workspace supports examination of an original circuit and a modified circuit after component replacement. The simulation interface includes transient, DC operating-point, and worst-case analysis options, together with duration and time-step controls and a results view. The paper documents the system concept, architecture, data representation, workflows, and prototype status. The current implementation is treated as a prototype: public deployment, comprehensive user evaluation, validated numerical accuracy, and complete persistent database operation have not yet been established. Accordingly, no quantitative performance or accuracy claims are made. The contribution is an integrated design that provides a basis for future verification, usability testing, and extension into a validated engineering tool.

***Keywords*- Electronic component selection, component comparison, alternative components, circuit simulation, web-based engineering tool.**

I. INTRODUCTION

Electronic circuit design involves more than selecting parts that perform the required function. A designer must also verify operating conditions, absolute and recommended electrical limits, package and pin compatibility, interface levels, current and power requirements, and the availability of suitable alternatives. Manufacturer datasheets remain essential sources for these decisions, but information is commonly spread across separate documents and distributor listings. Comparing candidate components manually can therefore be repetitive and can make it difficult to trace why one part was considered an acceptable substitute for another.

Circuit simulation provides a complementary way to reason about circuit behaviour before physical implementation. Established circuit simulators support analyses such as transient response and DC operating point, allowing a designer to inspect voltages and currents under defined models and conditions. However, the component research and replacement decision may still occur outside the simulation environment. This separation creates an opportunity for a workflow that connects component information, comparison, replacement, and circuit analysis.

This paper presents Component Selector and Circuit Simulator, a web-based engineering prototype intended to support this connected workflow. The system combines a structured component catalogue with comparison and alternative-finding functions and a circuit workspace in which an original design can be considered alongside a modified design. The project is intended for students, electronics hobbyists, and engineers who need a more organised way to investigate component choices.

II. LITERATURE SURVEY Electronic component selection is fundamentally a requirements-matching activity. A candidate part must satisfy the functional role of the original part and remain within the electrical, thermal, mechanical, and interface constraints of the application. Datasheets provide the authoritative limits and operating characteristics needed for this assessment. Because values may be specified under different test conditions, comparing isolated headline values without checking the associated conditions can lead to inappropriate substitution decisions.

Circuit simulation has long been used to analyse electronic networks. The SPICE family of programs established a widely adopted approach to numerical circuit analysis, including operating-point and time-domain analysis (Nagel, 1975). Simulation results depend on the circuit topology, component models, model parameters, solver configuration, and boundary conditions. Consequently, a simulator interface should make analysis type and time-related settings explicit and should avoid implying that a displayed result is experimentally verified unless it has been checked against a trusted reference.

Software engineering literature emphasises requirements analysis, modular design, verification, and iterative evaluation as important activities in developing dependable software systems (Sommerville, 2016; Pressman & Maxim, 2019). For a component-selection application, these principles imply that data quality, traceability, validation rules, and clear separation between catalogue information and computed simulation results are as important as the user interface.

The present project addresses the workflow integration problem rather than proposing a new circuit-solving method. It aims to make component data and comparison decisions available in the same application context as circuit modification and analysis. The prototype should therefore be understood as a software integration and workflow contribution; its engineering validity depends on future testing against authoritative datasheets and established simulation tools.

III. PROPOSED METHODOLOGY

3.1. Development approach and scope.

The system was developed iteratively as a web application. The available project history identifies a Next.js-based frontend and API-oriented backend development. A MYSQL-backed component and project database was part of the intended architecture, but complete dynamic persistence of component, project, circuit, and analysis records was still incomplete in the project discussions. The application is therefore described as a prototype with partially implemented persistence, not as a fully deployed production service.

The methodology followed a workflow-oriented design: identify the component-selection problem; define the component information required for comparison; organise selection and comparison functions; provide a circuit workspace for considering replacements; expose simulation-analysis controls; and present results for engineering interpretation. The implementation history also records a project workspace concept using an Arduino-based example containing an MQ-2 gas sensor, an LED, a buzzer, and a resistor. This example is used here only to explain the intended workflow, not to claim experimentally validated results.

3.2. Component Information Model

The component record is organised into descriptive and electrical groups. Descriptive fields include name, type, category, manufacturer, part number, functional role, and description. Electrical attributes are represented as optional values because not every attribute applies to every component. The known attribute set includes operating voltage, input voltage, output voltage, current rating, power rating, logic level, and frequency. This flexible structure permits components from different categories to be represented without forcing irrelevant fields to contain fabricated values.

Data group	Fields / examples	Purpose
Identity	Name, type, category, manufacturer, part number	Identify and classify the part
Functional description	Functional role, description	Explain what the component does
Electrical limits	Operating/input/output voltage; current and power ratings	Support constraint checking
Interface characteristics	Logic level, frequency	Support interface and signal comparison
Project context	Project, circuit, selected component, replacement	Relate component decisions to a design
Analysis context	Analysis type, duration, time step, results	Record simulation setup and outputs

3.3. Component Comparison and Alternative assessment

The comparison workflow begins with a selected component and a set of candidate components. The system is intended to display their descriptive and electrical attributes in a consistent comparison view. An alternative should not be treated as equivalent solely because its name or functional category is similar. A meaningful assessment must compare the parameters relevant to the application, including voltage range, current capability, power rating, logic compatibility, frequency where applicable, package or pinout, and operating conditions.

The prototype's comparison feature is intended to help users identify similarities and differences and to narrow the candidate list. Final suitability remains an engineering decision based on the original and replacement manufacturers' datasheets. Where a parameter is unavailable, it should be shown as unavailable rather than inferred. The project history does not establish a fully validated weighted-ranking algorithm; accordingly, this paper does not claim that alternatives are automatically certified as drop-in replacements.

3.4. Circuit Workspace and replacement workflow.

The intended project workflow provides a workspace containing a circuit representation and its component connections. A user selects a component, reviews potential alternatives, and applies a chosen replacement to create a modified circuit. The original and modified versions can then be considered side by side. In the example discussed during development, an Arduino-based circuit includes an MQ-2 sensor, LED, buzzer, and resistor. Replacing a part should trigger a review of the affected connections and electrical conditions before simulation.

This workflow is designed to preserve the relationship between a component decision and the circuit in which it is used. A robust implementation should retain the original design, record the replacement, and identify any changes to connections or parameter values. These are design requirements; the extent to which each is fully persisted in the current prototype requires verification against the running application and its database.

3.5. Simulation interface and analysis mode.

The simulation interface described in the project history supports an original-versus-replacement comparison and exposes three analysis choices: Transient, DC Operating Point, and Worst Case. It also provides duration and time-step settings, together with Run and Reset controls and a results display. Transient analysis is intended to examine time-varying circuit behaviour; DC operating-point analysis is intended to examine steady-state node voltages and branch currents; worst-case analysis is intended to explore behaviour under specified parameter variations or limiting conditions.

The availability of these controls in the interface should not be confused with proof that a standards-compliant numerical solver is integrated or that all displayed metrics are calculated from validated circuit models. Earlier project discussions explicitly described displayed demo metrics. Before publication as a validated simulator, the implementation must document the solver or external simulation engine, supported model formats, numerical methods, convergence handling, and validation against known circuits or a reference simulator.

3.6. Technology and system Architecture.

The project history identifies a Next.js frontend and a separate API backend used during development. A MySQL database was planned for component and project data. The logical architecture can be divided into a presentation layer, application/API layer, component-data layer, circuit-workspace layer, and simulation/analysis layer. The presentation layer handles user interaction; the API layer mediates application operations; the data layer stores component and project records; the workspace layer maintains circuit structure and replacement context; and the analysis layer manages simulation configuration and result presentation.

Layer	Responsibility	Status evidenced in project history
Presentation	Component search, comparison, project workspace, simulation controls	Frontend under development using Next.js
Application/API	Authentication and application operations	API-based backend development evidenced
Data management	Component, project, circuit and analysis persistence	MySQL planned; full dynamic persistence incomplete
Circuit workspace	Original circuit and replacement circuit context	Workflow and UI described; verify completeness
Simulation/analysis	Transient, DC operating point, worst-case options and results	UI controls and demo metrics described; solver validation not established

IV. RESULT

The prototype provides an interface for exploring electronic components, examining their electrical specifications, comparing different components, and identifying potential alternatives. It also includes a circuit simulation workspace designed to supportthe analysis of original and modified circuit configurations.

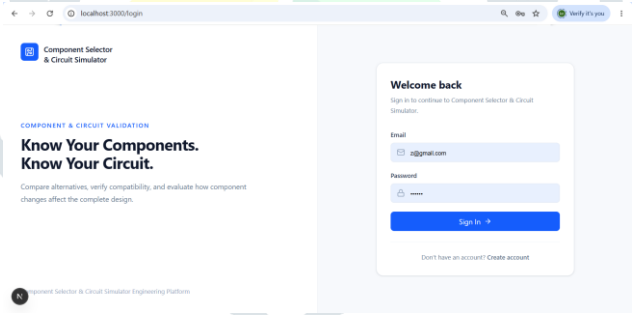

Figure 1. Login Page

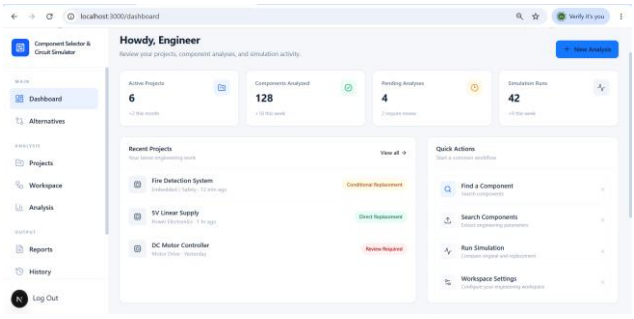

Figure 2. Dashboard Page.

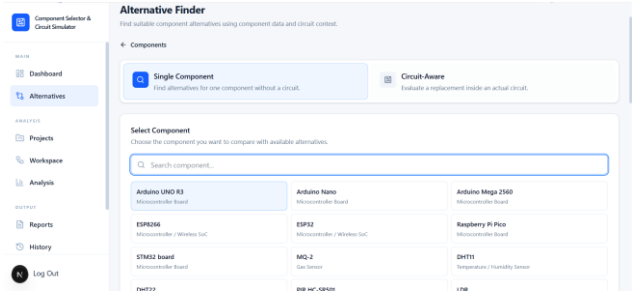

Figure 3. Alternative Finder Page.

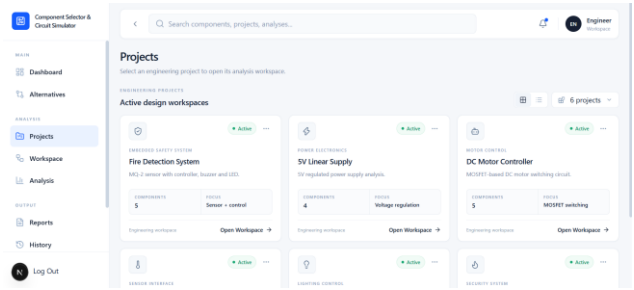

Figure 4. Project Page.

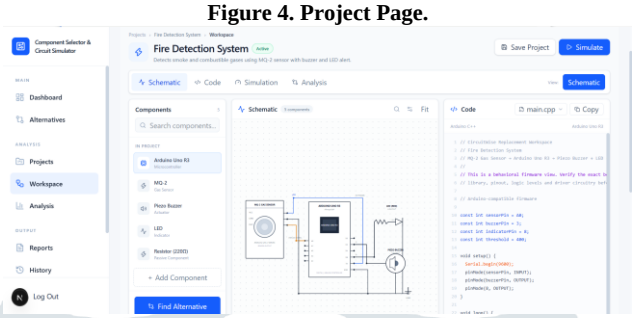

Figure 5. Project- Fire Detection System.



Figure 6. Analysis Results.

V. DISCUSSION:

Integrating component selection with circuit analysis can reduce the fragmentation of the design workflow. A user can begin with a part-level question, inspect alternatives, and then consider the proposed change in its circuit context. This is particularly useful in educational settings, where learners may need to understand not only whether two components share a functional role but also why their electrical ratings and interface characteristics matter.

The principal engineering challenge is data correctness. Component specifications should be linked to manufacturer datasheets or other authoritative sources, and the system should retain source, revision, and retrieval information. Parameters should be compared only when their definitions and test conditions are compatible. The application should distinguish absolute maximum ratings from recommended operating conditions and should avoid presenting a candidate as safe merely because a small subset of its specifications matches.

A second challenge is simulation validity. A simulation result is meaningful only when the circuit topology, device models, parameters, source conditions, and solver settings are appropriate. The interface's three analysis modes provide a useful organisation for the workflow, but numerical results must be validated using benchmark circuits and comparison with a recognised simulation engine. Worst-case analysis in particular requires a clearly specified tolerance model and method for selecting parameter extremes.

A third challenge concerns software completeness and reliability. The incomplete persistent database functionality noted during development limits the ability to claim durable project histories or reproducible analysis records. Before production use, the system should implement database migrations, validation, authentication and authorisation controls, error handling, backup procedures, and audit trails. The web application should also be deployed to a stable test environment so that repeatable functional and performance testing can be conducted.

VI. CONCLUSION:

This paper has described the design and development status of Component Selector and Circuit Simulator, a web-based prototype intended to connect electronic component discovery, specification comparison, alternative exploration, circuit replacement, and simulation-oriented analysis. Its component data model groups identity, functional description, and optional electrical attributes, while its workflow links a selected component to a modified circuit and provides transient, DC operating-point, and worst-case analysis controls. The project offers a coherent foundation for an integrated component-selection and circuit-analysis environment. However, it remains a prototype: public deployment, complete persistent data management, numerical solver validation, and formal user evaluation have not been established. Completing these activities is necessary before making claims about engineering accuracy, reliability, or practical effectiveness.

VII. ACKNOWLEDGMENT

We would like to express my sincere gratitude to project guide Prof. S.A. Khusro, for her invaluable guidance and support throughout the project on “Component Selector and Circuit Simulation”. Her expertise and mentorship have been instrumental in shaping the content and direction of this project

REFERENCES

[1] Chatzipetrou, P., Papatheocharous, E., Wnuk, K., Borg, M., Alégroth, E., & Gorschek, T. (2020). Component attributes and their importance in decisions and component selection. Software Quality Journal, 28, 567–593. https://doi.org/10.1007/s11219-019-09465-2

[2] Fernández-García, A. J., Iribarne, L., Corral, A., Criado, J., & Wang, J. Z. (2019). A recommender system for component-based applications using machine learning techniques. Knowledge-Based Systems, 164, 68–84. https://doi.org/10.1016/j.knosys.2018.10.019.

[3] Gholamshahi, S., & Hasheminejad, S. M. H. (2019). Software component identification and selection: A research review. Software: Practice and Experience, 49(1), 40–69. https://doi.org/10.1002/spe.2656