



Personal Personal Project Cross-Platform Framework Research

Michał Raczkowski

OL S6

4465024

Contents

1	Overview	2
2	Research	2
2.1	Electron.js	2
2.2	Qt	2
2.3	Flutter	2
2.4	NW.js	2
3	Comparison table	3
4	Conclusion	3
4.1	Extensive GUI Customization	3
4.2	Access to System Resources	4
4.3	Cross-Platform Support	4
4.4	Familiarity with JavaScript/Typescript	4
4.5	Active Community and Abundant Resources	4
4.6	Ecosystem Maturity	4

Version	Date	Author	Comment
0.1v	03.06.23	M. Raczkowski	Overview, Research, Comparison table
0.2v	04.06.23	M. Raczkowski	Conclusion

1 Overview

This research conducted a comparison of several frameworks, including Electron.js, Qt, Flutter, and NW.js, for developing the ArduFlow application. The evaluation focused on factors such as GUI customization options, resource utilization, UI rendering, cross-platform capabilities, and available libraries and tools. Electron.js emerged as a framework that allows developers to build cross-platform desktop applications using web technologies, providing extensive GUI customization and direct access to system resources. Qt offers a native look and feel with efficient resource utilization, while Flutter provides fast UI rendering and a single codebase for multiple platforms. NW.js combines web technologies with Node.js and system access. The selection of the most suitable framework depends on specific requirements such as GUI customization needs, performance considerations, development team expertise, and the availability of libraries and tools.

2 Research

This comparison examines various frameworks for developing the ArduFlow application. The evaluation aims to identify the most suitable framework for building the LED 8x8 matrix editor. Factors such as GUI customization, performance, platform compatibility, and available tools will be considered. The objective is to choose the optimal framework that aligns with the specific requirements of the ArduFlow project.

2.1 Electron.js

Electron.js is favored for its familiarity with web technologies, extensive GUI customization options, and direct access to system resources. However, it may have higher memory consumption and larger application size.

2.2 Qt

Qt stands out with its native look and feel, high performance, and efficient resource utilization. It offers extensive GUI customization options using QML. However, it has a steeper learning curve and limited availability of existing libraries.

2.3 Flutter

Flutter provides fast and visually appealing UI rendering, a customizable framework, and a single codebase for multiple platforms. However, it is relatively newer with a smaller community and limited availability of platform-specific libraries.

2.4 NW.js

NW.js combines web technologies with Node.js and offers access to a wide range of Node.js modules and native UI components. However, it may have higher memory consumption and larger application size.

3 Comparison table

Frame- work	Pros	Cons	Language	UI Frame- works Avail- ability	Complexity of Development
Elec- tron.js	Extensive GUI customization options	Higher memory consumption	JavaScript/ Typescript	Web-based frameworks (e.g., React, Angular)	Moderate
	Cross-platform support	Larger application size			
	Access to system resources	Potential performance limitations			
Qt	Native look and feel	Steeper learning curve	C++	Qt QML	High
	High performance and resource utilization	Limited availability of libraries			
	Extensive GUI customization options				
Flutter	Fast and visually appealing UI rendering	Relatively newer with a smaller community	Dart	Flutter	Low
	Single codebase for multiple platforms	Limited availability of platform-specific libraries			
	Hot reload feature for quick iterations				
NW.js	Extensive GUI customization options	Higher memory consumption	JavaScript/ Typescript	(e.g., React, Angular)	Moderate
	Access to Node.js modules and APIs	Larger application size			

4 Conclusion

Based on the functionalities and features required for the ArduFlow application mention in "Personal Project Idea, Specification and Requirements", the best solution is Electron.js.

4.1 Extensive GUI Customization

Electron.js allows for highly customizable graphical user interfaces (GUIs) using web technologies like HTML, CSS, and JavaScript. This flexibility enables the creation of a tailored and

visually appealing interface for the LED 8x8 matrix editor.

4.2 Access to System Resources

Electron.js provides direct access to system resources, allowing seamless communication with the Arduino board to control the LED matrix. This enables real-time interaction and synchronization between the application and the hardware.

4.3 Cross-Platform Support

Electron.js offers cross-platform compatibility, allowing the ArduFlow application to run smoothly on multiple operating systems, including Windows, macOS, and Linux. This broadens the reach of the application and ensures usability across different devices.

4.4 Familiarity with JavaScript/Typescript

Electron.js utilizes JavaScript/Typescript, a widely adopted programming language, which is familiar to a large number of developers. This familiarity simplifies the development process and facilitates the utilization of existing JavaScript/Typescript libraries and frameworks.

4.5 Active Community and Abundant Resources

Electron.js benefits from a thriving and active community of developers. This vibrant community provides access to extensive documentation, libraries, and resources, making it easier to find solutions, seek guidance, and address any development challenges encountered during the ArduFlow project.

4.6 Ecosystem Maturity

Electron.js has been widely adopted in the industry and has proven its stability and reliability through the development of numerous successful applications. Its ecosystem is mature, offering a wide range of plugins, libraries, and tools that enhance development efficiency and provide additional functionality.