Development Toolchain Comparison

The PIQUE Smart Display Module is preconfigured for front-end application development using any of these toolchains:

- Electron – a browser-based front end using HTML, CSS, and Node.js technologies
- Qt– a native-experience cross-platform widget toolkit using C++ and QML.
- LVGL – a lightweight graphics library using C/C++

Electron

The Electron framework uses the Chromium browser rendering engine with a Node.js back-end to create a naturally cross-platform application. Because the application front-end is rendered using HTML, embedded images/fonts, and CSS, design and programming assets can be shared between Electron apps and web pages. Node.js provides a large ecosystem of code libraries to speed development.

Qt

Qt is a popular and long-lived GUI development system. Qt uses a library of native widgets written in C++ along with a proprietary design language, QML. Qt runs on desktop operating systems and a wide variety of embedded platforms.

LVGL

LVGL is a lightweight embedded GUI development library that prioritizes low memory footprint. Applications code is developed using C/C++ or Micropython. While C/C++ is rarely used for desktop development, the small memory footprint and simple driver interface enable scaling to very small processors and screen sizes.

Usage

To use this guide, simply identify the characteristics that are a priority for your application and select a framework that offers the advantages in those categories. This document is intended only as a starting point for decision-making, as some of these measurements are subjective or may change for a certain application.

Tablet-Like Touchscreens for the Raspberry Pi Compute Module 3+

Development Time

Rapid application development in GUI applications enables you to quickly respond to customer input, accommodate requirements changes, update the customer experience, and get a product to market quickly. The large ecosystem of web development tools and their ease-of-use makes Electron is a clear winner in this category. Qt does perform well in this category, though, as Qt offers a form/widget designer and code generator. LVGL will earn some points here when the code generator (Edgeline) becomes available.

	long								short	
Electron									■	■
Qt					■	■				
LVGL	■	■								

Native Font and Image Support

Related to Development Time, Electron and Qt provide the most support for rendering images and fonts without changes. LVGL requires the use of a tool to convert font and images into raster objects stored in code.

	all								none	
Electron	■	■								
Qt		■	■							
LVGL									■	■

Resource Use

Electron, because it's based on the Chromium browser, has a reputation for high memory usage. It's important to realize, however, that this is done in order to maximize performance, and memory is released to other applications that request it. Except in specific circumstances, this should not be a critical factor.

	low								high	
Electron									■	■
Qt		■	■							
LVGL	■	■								

Boot Time

In high uptime applications, a relatively time from powerup until the application is available is tolerable. In other applications, the user expects very quick access to machine controls, etc. Because Electron uses high resources and relies on the Linux windowing system to load, Qt and LVGL have a large advantage here.

	short								long	
Electron									■	■
Qt		■	■							
LVGL	■	■								

Popularity

Why do we care if a tool is popular? Popular frameworks benefit from a large base of user feedback, support, and contributed tools. Using mainstream, popular tools can also make locating and employing talented developers easier. Electron is very popular, but because of its longevity, Qt also enjoys widespread worldwide support.

	least								most	
Electron									■	■
Qt							■	■		
LVGL	■	■								

Cross-Platform Development

Every team is looking to leverage its developed assets as widely as possible. Electron may be helpful in organizations that already use web container apps or Electron desktop programs. Qt may be ideal for organizations planning to support mobile apps, desktop apps, and PIQUE going forward. Because the popularity of C/C++ for desktop/mobile development has waned, LVGL lags in this category.

	least								most	
Electron									■	■
Qt									■	■
LVGL	■	■								

Video and Animations

Playing full-motion video is a must-have in some applications. In others, a lower-resolution or frame-limited video is adequate. Electron and Qt (used properly) both provide access to the PIQUE's graphics accelerator (GPU) for smooth video up to full-HD at 60 FPS. LVGL is capable of lower-end video and is still provides snappy animations and screen transitions.

	worst								best	
Electron									■	■
Qt									■	■
LVGL	■	■								

Licensing

LVGL and Electron are both free, open-source frameworks available under the very permissive MIT License. While Qt is available as an open-source project under the GPL license, purchase of a license is required for most commercial applications. Contact The Qt Company (qt.io) for details.

	open								licensed	
Electron	■	■								
Qt									■	■
LVGL	■	■								

Cost-Reduction Opportunities

After product viability has been proven, code churn has slowed, and the market opportunity is identified, many companies look to cost-reduce the hardware design. While it allows for very quick initial application development, Electron-based applications will require significant re-coding to run on cost-reduced hardware. The Circuit Foundry's minimized platforms support Qt and LVGL, making cost-reduction simpler.

	easy								difficult	
Electron									■	■
Qt			■	■						
LVGL	■	■								

Conclusion

This document should guide your first steps toward selection of your development toolchain. Because this selection can determine the success or failure of your product, it's important to solicit input from all available stakeholders, with or without aid of this document.

Resist the urge to overly "score" or "quantify" the options. If your choice doesn't easily stand out from the others, consider testing development of a small part (e.g. main screen) of your GUI using both tools, then compare them collectively after the exercise is complete.

	Electron	Qt	LVGL
Fast Development	✓✓	✓	
Native Font and Image Support	✓	✓	
Low Resource Use		✓	✓✓
Fast Booting		✓	✓
Popular/Mainstream	✓✓	✓	
Cross-Platform	✓	✓	
Video and Animation	✓	✓	
License	MIT Open Source	Commercial ($)	MIT Open Source
Easy Cost-Reduction		✓	✓✓