

# Luis A. Rodriguez-Condit

Technical Architect – Embedded & Web Systems

Mobile: +1-972-850-6310

Email: [larc63@gmail.com](mailto:larc63@gmail.com)

Linkedin: [www.linkedin.com/in/larc63](http://www.linkedin.com/in/larc63)

github: [www.github.com/in/larc63](http://www.github.com/in/larc63)

Location: Wylie, TX

## Summary

I'm a Senior Software Engineer with over 20 years of experience building software that ranges from video games for small resource limited mobile devices to web applications used by millions of users. I prioritize user experiences to help them achieve whatever their trying to do in a simpler, quicker fashion. I am capable of being an individual contributor, or leading teams where I use my organizational and emotional skills to deliver what the customer or business needs. In a mentor or lead role, I tend to be empathetic, looking for ways to keep the team's morale up so we can all be more efficient. Overall I love to design, write and deliver software that is not only technically sound, but also meaningful and user-centered.

## Skills

Languages	C, JavaScript, TypeScript, CSS, HTML, Java, C++, Python, Objective-C, Swift, z80 Assembly, Brew, J2ME
System and low level	Emscripten, Markdown, Hardware Emulation, WebAssembly, OpenAPI/Swagger, Java Native Interface, WebSerial, WebUSB, USB Communication, Cross-Compilation
Mobile/Web/Frameworks	Angular, Spring Boot, Lua, Gulp, Ant, Mocha, Jasmine, Selenium WebDriver, Jest, React Testing Library
Leadership	Test-Driven Development, Technical Leadership, Agile/Scrum, Project Coordination, Git

## Transferrable Skills

Technical Architecture	Full-Stack Development, System Integration, Low-Level Bridging, Cross-Compilation, Architected and implemented solutions that bridge proprietary embedded systems (C/z80) with modern web applications (Angular/TypeScript) via WebAssembly and Emscripten.
Web/Front-End Expertise	Angular, TypeScript, Responsive Design, API Integration (RESTful), Designed and implemented Angular applications for calculator emulation, license activation, and user utilities, utilizing third-party SSO and communicating with Java/SpringBoot middleware.
Embedded & Systems	Hardware Emulation, Low-Level Programming (C, z80), Communication Protocols (USB, WebSerial/WebUSB, JNI), Designed the glue layer for targeting embedded code on the web. Established cross-platform USB communication architecture using abstraction layers and JNI.
DevOps & Automation	Build System Optimization, CI/CD, Scripting (Gulp, Node.js, Jenkins), Deployment, Developed and maintained complex build scripts, optimizing workflow into parallel Jenkins jobs, resulting in a 70% build time speed-up.

Technical Leadership	Scrum Master, Technical Requirements, Cross-Functional Coordination, Code Review, Led development teams, serving as Scrum Master and Development Lead. Authored technical requirements based on user needs. Coordinated business-wide tech sharing sessions.
Mentorship & Team Morale	Empathetic Leadership, Mentoring, Knowledge Transfer, Managed and mentored engineering teams, offering tutorials and code deep-dives to improve team proficiency and maintain morale.
Software Quality & Testing	Test-Driven Development (TDD), Automation Frameworks, Code Quality Metrics, Prototyped automation frameworks (Node.js, Mocha, Selenium WebDriver). Designed and met code quality standards (unit testing, static analyzers, mandatory code reviews).
Product & User Focus	User-Centered Design, Accessibility Implementation, Requirements Analysis, Prioritized user experiences for simpler, quicker solutions. Implemented code in z80 and JavaScript to enable screen reader accessibility for calculator emulators.
Risk & Compliance	Open Source Auditing, Security Remediation, Responsible for reviewing third-party library lists to ensure eligibility and manage licensing. Led teams to address IT security scan concerns (modifying HTTP headers).

## Work Experience

### **Texas Instruments -- Software Architect**

Tenure involved an evolution from rapid functional prototyping (building POCs and starter code, thus, reducing time-to-market) and deep systems engineering to technical leadership roles and architectural oversight. Considered the go-to person for bringing low-level hardware emulation and proprietary communication protocols to desktop and web applications. As part of the metrics task force, designed and met code quality metrics with tools like unit tests, code sanitizers, linters and static analyzers. This career progression was formally recognized with an appointment to the first rung of TI's Technical Ladder.

#### **Open Source Auditing**

**2018-2025**

Responsible for reviewing third party library lists to determine their eligibility for integrating into our products. End result of this per-release activity is a list of attributions that would get appended to the EULA for each shipped product.

#### **Tech Sync Coordinator**

**2018-2025**

Coordinated business-wide information sharing sessions and recruited presenters for the same. This weekly series resulted in broader understanding of deep technical topics as well as business-related subjects.

#### **Calculator Web Apps**

**2025**

Led team that implemented the functionality for 2 web apps: One for USB connectivity with the calculator and another shell for the emulator of the calculator. Serving functions of development lead, scrum master. Reviewed user requirements and wrote technical requirements. Web apps were written in Angular, with libraries compiled from C via Emscripten. For USB connectivity, used WebSerial API.

#### **Calculator File Transfer Library**

**2024-2025**

Designed and implemented functionality for performing file transfers between 1) Web app and physical calculator using WebSerial API and 2) A webapp that contains an emulated calculator, making the API between both libraries match so they can be a drop-in replacement for each other

<b>Calculator Transformation Graphing</b>	<b>2024-2025</b>
Implemented functionality to support animating graphs on new hardware platform written in C, using the APIs provided by the calculator's tools.	
<b>Calculator Build Environment</b>	<b>2020-2025</b>
Developed and maintained gulp.js scripts that automated the complex process of invoking disparate platform build tools, such as CMake and MSBuild. This build system was later optimized for speed by restructuring the workflow into parallel Jenkins jobs, resulting in a 70% build time speed-up. The task division was actively managed and rebalanced to sustain peak performance.	
<b>Calculator Device Emulation</b>	<b>2020-2025</b>
Designed and implemented the glue layer for targeting embedded code on the web by transpiling to WebAssembly. Coordinated cross-functional team that did development and test for new features. Designed and implemented an automation entry point using WebSockets.	
<b>TI-84 Plus CE Online Calculator</b>	<b>2022</b>
Implemented Angular application that provides access to an emulator for the TI 84 Plus CE calculator on web browsers. Implemented single-sign-on usage for user authentication (OAuth identity) and license validation (entitlement) via 3rd party provider. This was the first publicly available TI calculator on a web app with *over 1500 users in first month.	
<b>TI-Nspire Online Calculator</b>	
Consulted with team implementing the glue layer as well as the application in Angular. Implemented prototype for in-emulator clipboard interaction with system clipboard using the Clipboard API.	
<b>License Activation Center</b>	<b>2020-2021</b>
Designed and implemented a key validation interface in coordination with our DBA to provide users a means to activate license codes. Front end implementation written in Angular with a RESTful middleware implemented in Java+SpringBoot communicating with an Oracle database. Implemented multi-language support using Angular's i18n tools and custom scripts to convert content into formats compatible with the translation workflow.	
<b>TI-84 Plus CE Chrome App</b>	<b>2020</b>
In reaction to the covid pandemic, led the team that worked to support remote learning on Chrome OS allowing students coming back from spring break to have a TI calculator to work with. To date, the application has been used by over 3.5 million users. -- written in angular reusing previous work from the ExamCalc product.	
<b>Automated Test Framework Prototype</b>	<b>2019</b>
Prototyped automation framework for interacting with calculator libraries in the browser using Node.js, CucumberJS, Mocha and Selenium Web Driver.	
<b>Web Apps Server-Side</b>	<b>2019</b>
Led team in determining Tomcat and Apache Web server configuration changes needed to address concerns brought up by IT security scans. Changes included modifying HTTP headers and additions to meta tags in the web app's HTML	
<b>TI-Nspire Connect</b>	<b>2019</b>
Designed and implemented Angular shell to run the web app, as well as the JavaScript and C glue layers for cross-compiling NavNet, the TI-Nspire line of calculator's communication library with Emscripten to enable communicating with a TI-Nspire CX II device via WebUSB. 28k on it's first year.	
<b>TI-83 Plus CE ElectronJS App</b>	<b>2019</b>

Designed and implemented ElectronJS shell to run an emulator of the TI-83 Plus CE calculator allowing teachers completing France's certificate of aptitude for secondary school teaches (CAPES). Work for this application also involved investigations into

---

#### ExamCalc Accessibility

2017-2018

Implemented code needed in both z80 assembly and javascript to support market needs to make calculator emulators interact correctly with a screen reader for the vision impaired. Due to the nature of the calculator emulators, no `_aria_` element is readily available, so the emulator needs to provide the messages to be fed to a screen reader.

---

#### TI Connect For Chrome OS

2015-2016

Ported the communications layer on a utility that talks to TI calculators using Chrome's USB API. Implemented user-facing features in AngularJS in combination with Chrome APIs.

---

#### TI Connect CE

2013-2015

Delivered cross-platform compatibility for a proprietary CARS communication protocol by porting the codebase to generate platform-native libraries (Windows/macOS). Established the USB communication architecture using an abstraction layer and integrated the functionality into the application via Java Native Interface (JNI).

---

#### TI SmartView (Hardware Emulation)

2013-2016

Wrote and reviewed user requirements. Designed and implemented features related to interacting with emulated hardware. Helped the development team solve memory and performance issues.

---

#### TI 84 Plus CE ExamCalc

2013

Implemented emulators for ez80-based devices which have gone through our client's pilots involving 800k users without any reported issues.

---

#### ExamCalcs TI-84 Plus SE, TI-84 Plus C

2012-2013

TI calculators that are approved for use on \*ACT®, SAT®, and AP® exams\*, as well as most \*state assessments.\* . Ported and implemented calculator emulators to the web. Adapted z80 opcodes and emulated the TI-84 Silver Edition and TI-84 Plus C's ASICs. Ported Toshiba T4 cpu emulation from Java and implemented the TI-108 and TI-30MV's ASICs. Pilots involving 800k users without any reported issues.

---

#### ExamCalcs

2012-2013

TI calculators that are approved for use on \*ACT®, SAT®, and AP® exams\*, as well as most \*state assessments.\* . Ported and implemented calculator emulators to the web. Adapted z80 opcodes and emulated the TI-84 Silver Edition and TI-84 Plus C's ASICs. Ported Toshiba T4 cpu emulation from Java and implemented the TI-108 and TI-30MV's ASICs. Pilots involving 800k users without any reported issues.

---

#### Document Display Standardization

2012

Given the strong desire for documents to display with the same word-wrapping and character counts on all supported devices (iPad being a new addition): researched and developed guidelines and layout rules for document rendering when being displayed on different screen resolutions.

---

#### Lua Glue

2011

Designed and implemented the infrastructure allowing new classroom activities to use the Lua scripting language. The C counterpart for the Lua bindings, as well as the Lua engine itself gets cross-compiled to run on different platforms including iOS, Nucleus and desktop operating systems.

---

#### Image Question type

Designed and implemented some of the infrastructure necessary to create new classroom activities using the Lua scripting language. The C counterpart for the Lua bindings, as well as the Lua engine itself gets cross-

compiled to run on different platforms including iOS, Nucleus and desktop operating systems. The use of Lua would open up the possibility of expanding TI-Nspire's question types in the future without needing to write them in Java or C

---

#### **UI And Background Thread Refactor**

Worked in conjunction with the development team to refactor the usage of JNI by piping requests to the backing C libraries into a single thread, minimizing UI lockups and improving performance and memory usage overall.

---

#### **Digital Chocolate -- Game Engineering Manager**

Managed the team doing porting for the US carriers' devices.

##### **Game Engineering Manager**

**2009-2010**

Coordinating project planning for various carriers in the US, working in conjunction with the sales and marketing teams to reduce production costs and improve revenue. Lead and mentor the engineering team, offering tutorials and code deep-dives where needed.

---

##### **Brew In App Purchase Library**

**2010**

Created client module for Qualcomm's Application Value Billing API (C/C++) in conjunction with the team in Bangalore, India. Allowing games to have in-app-purchases as well as promoting code reuse on future game titles

---

##### **Star Invasion**

**2010**

Supported team that created Alien Invaders by porting for small-screened and resource limited devices. Aided the creation team to reduce the footprint for their project; done in J2ME, targeting small-screened and resource limited devices.

---

##### **Rollercoaster Rush for iPhone**

**2010**

Ported iPhone implementation of OpenGL ES game to Qualcomm's Brew framework

---

##### **LemonQuest -- iPhone Lead Programmer**

Developing games and applications for the iPhone OS. Coordinated development with a geographically distributed team. Prompted the company to implement updates to infrastructure to improve productivity.

##### **Tool Programmer**

**2008**

Aided in the development of a stage editor for a racing game, done in Java.

---

##### **Circulate and Circulate Prologue**

**2008**

Originally a PC based game; it was ported to the iPhone platform. Game used every user-friendly feature on the iPhone, including it's 3d sound capabilities with Open AL and it's accelerometer. Ported the DirectX implementation to the iPhone's OpenGL ES, converting 3d models from quads to triangles on the. Translated levels originally designed for desktop/mouse to be usable on a touch device. This game earned a bronze medal from pocketgamer.co.uk.

---

##### **iFit**

**2009**

Wrote and application that provided pedometer functionality as well as fitness tips for the iPhone

---

##### **Gameloft -- Lead Game Programmer**

Various roles, starting from Porting Engineer, Lead Developer, 3d Programmer and R&D Programmer

##### **Blockbreaker for iPhone**

**2007-2008**

Developed and ported first prototype for the brick-breaker game using the Celestial framework for iPhone. The Mexico studio's team was the first to build a successful prototype which led to the studio being assigned more

iPhone projects. Used C-implementation for OpenGL ES as graphics engine and sourced models from other platforms.

---

**Bikini Volleyball Mini Games** 2007

Implemented mini-games that were included with the main volleyball game when built for devices that would support them

---

**3d Programmer** 2006-2007

In charge of porting multiple game titles for resource-limited, but 3d-capable devices. All ports used Java's JSR184 3d implementation for mobile devices and were adapted by tweaking frustum setup and removing bottlenecks in graphics pipelines where possible, resulting in improved performance or, in some more extreme cases reducing models' polygon count to achieve acceptable frames-per-second and keep games playable

---

**Virgin Mobile Team Lead** 2005-2006

Led team charged with porting games to Virgin Mobile's catalog of space-constrained devices, often needing to reduce games' scope and swap out graphics with lower-detailed versions in addition to performing micro-optimizations to improve performance

---

## Other Projects

**50 Chinese Words** 2009

Wrote an application that reproduces the strokes necessary for writing chinese characters as well as playing their sound. Developed a Chinese character writing tutor for the iPhone.

**Big 2 Poker** 2010

Wrote graphics engine using C and OpenGL ES. Created art for a Big 2 Poker card game for iPhone OS.

**iAarti for Katha Films** 2010

Wrote a religious-focused application that shows images, plays sounds and uses the iPhone's accelerometer to interact with the user.

## Education

**Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)**

Completed coursework towards: Master's in Electronics and Telecommunications with specialization in control systems and automation, 2002-2005

## Academic Project

**Robot Control Over Ethernet** 2004

Developed RT-Linux kernel modules to communicate a PC that performed the PD control algorithm and a driver PC that sent control signals and motor positions over TCP sockets with low-latency, minimal-jitter to control a 2-degree-of-freedom robot arm. Control PC had a Java Swing UI connected to the kernel module via Java Native Interface. The driver PC performed I/O operations directly to a data acquisition card from kernel space.

**Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM)**

Bachelor's in Electronics Systems Engineering, 1997-2001

## Academic Project

**Control And Automation Seminar** 2021

Attended summer student exchange at the Université de Poitiers, France with a focus on control systems engineering.

## Professional Development

- Learn SQL - [Codecademy](#) - 2026
- Claude Code In Action - [Skilljar+Anthropic](#) - 2026

- Learn Redux - [Codecademy](#) - 2025
- AWS Cloud Practitioner Essentials - [AWS skill Builder](#) - 2025
- Next.js Fundamentals - [Codecademy](#) - 2025
- OpenAPI Fundamentals (LFEL1011) - [Linux Foundation](#) - 2025
- Learn React Testing - [Codecademy](#) - 2025
- Learn React: Lifecycle Methods - [Codecademy](#) - 2025
- Learn React - [Codecademy](#) - 2025
- Bluetooth Low Energy Fundamentals - [Nordic Developer Academy](#) - 2025
- A Beginner's Guide to Linux Kernel Development (LFD103) - [Linux Foundation](#) - 2025
- Learn C++ - [Codecademy](#) - 2025
- iOS developer nanodegree - [Udacity](#) - 2015
- Front-end web developer nanodegree - [Udacity](#) - 2014-2015
- Website Performance Optimization - [Udacity](#) - 2014
- Object-Oriented JavaScript - [Udacity](#) - 2014
- Javascript Basics - [Udacity](#) - 2014
- Intro to HTML and CSS - [Udacity](#) - 2014
- HTML5 Game Development - [Udacity](#) - 2014
- jQuery - [Codecademy](#) - 2013
- Software Testing - [Udacity](#) - 2012
- Web Application Engineering - [Udacity](#) - 2012
- Artificial Intelligence for Robotics - [Udacity](#) - 2012
- Introduction to Artificial Intelligence - [Stanford](#) - 2011

DISCLAIMER: Portions of this document were created using an AI to tailor experience points to the target role. All content, including dates, titles, and technical claims, is 100% accurate and based on my verified career history. I have thoroughly proofread and confirmed its factual correctness.