



WEBINAR
EMBARCADOS

A EVOLUÇÃO DO ECOSSISTEMA DA **LVGL** PARA SISTEMA EMBARCADOS



mouser.com

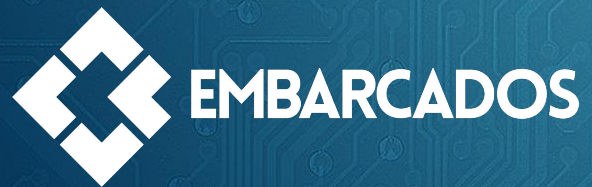


HALYSSON CARVALHO

Engenheiro de Sistemas Embarcados

A Evolução do Ecossistema da LVGL para Sistemas Embarcados





mouser.com

Agenda

- Journey
- Ecosystem Overview
- LVGL Open
- LVGL Pro
- Unlock your UI project
- Tools and Channels
- EV Demo

Journey



2016

Origin

Side project by
Gábor Kiss-Vámosi

2019-2020

v6 / v7

UI system, style
cache & real-time
fonts

2021

Launched v8

Automatic alignment,
flex/grid layouts

2021-2023

Partnerships

LVGL Kft, NXP, Arm,
Xiaomi; 400+
contributors

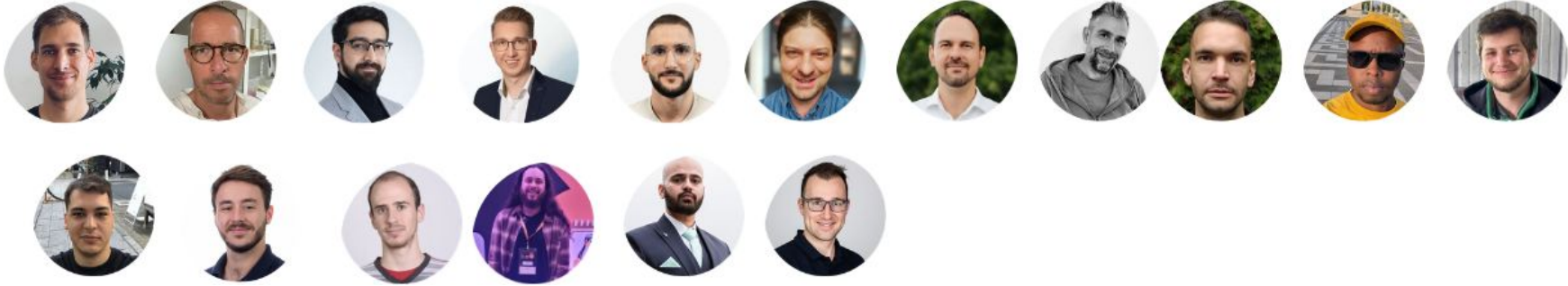
2025-2026

Ecosystem

LVGL Pro Editor,
Figma integration,
3D & MPU support

About LVGL

Two entities in the UK and Hungary, with a team distributed across six countries.



Ecosystem Overview

One ecosystem covering the entire embedded-UI lifecycle, from prototype, to production, to safety-critical.

Runtime

LVGL Open

Open-source UI runtime.
2D, 2.5D, 3D, MCU to MPU.

LVGL Safe

Certifiable runtime for
safety-critical UIs.

Tooling

LVGL Pro

XML editor, live preview,
drag-and-drop, and CLI.

Services

UI DEV & SUPPORT

Implementation, integration, and
expert support.

STRATEGIC PARTNERSHIPS

Chipmakers • (RT)OS • Display • Distribution • Service providers

LVGL Open

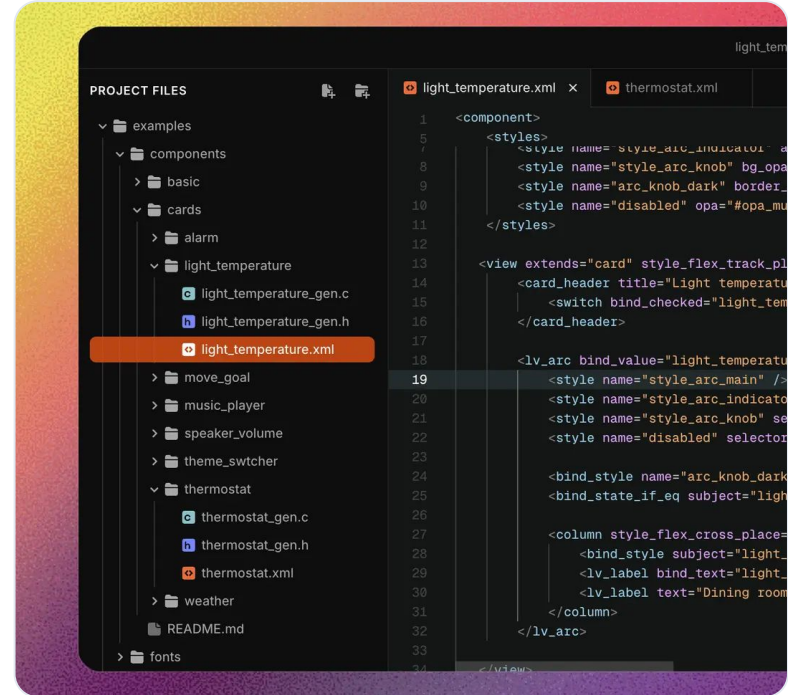
- Fully **open-source** on GitHub, free for commercial use
- Full-featured UI library for smartphone-like consumer UIs
- **Scalable** from bare-metal MCUs to MPUs with Open GL/3D
- Downloaded somewhere in the world every 5 seconds



LVGL Pro

A UI Editor to build UIs at lightning speed and collaborate smoothly.

- Editor **generates code** based on XML, **drag-and-drop** interface and Command Line Interface (CLI)
- Supports the **Model Context Protocol**, allowing you to work with **your favorite AI**.
- **Online Preview**, instantly preview and share your UI online with collaborators.



LVGL Pro - Licenses

No royalties, no per-seat fees, no fine print. From personal projects to enterprise-wide standardization.

Community

For makers, personal and open source projects

Free

Product

For smaller OEMs with few products

\$20,000
per product

Platform

For large-scale enterprise with floating licenses, on-premise license, servers and more

Custom

Unlock your UI Project

Component

The core building block for creating new UI elements in your project.

New Component

Create custom, specialized elements to tailor the interface directly to your design requirements.

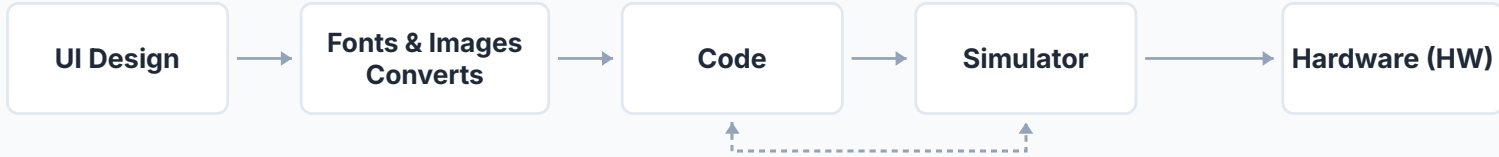
Screen

Organize layout flows inside self-contained screens that manage structures and animations.

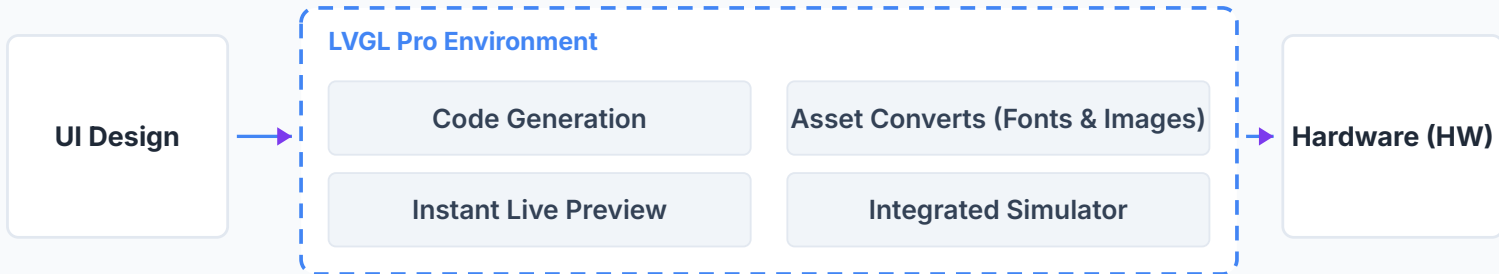
Widget

Deploy fundamental, highly reusable UI elements built to cooperate cleanly across components.

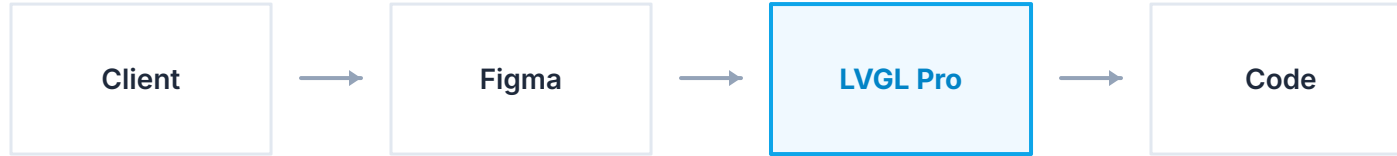
Traditional Workflow (**Without** Editor)



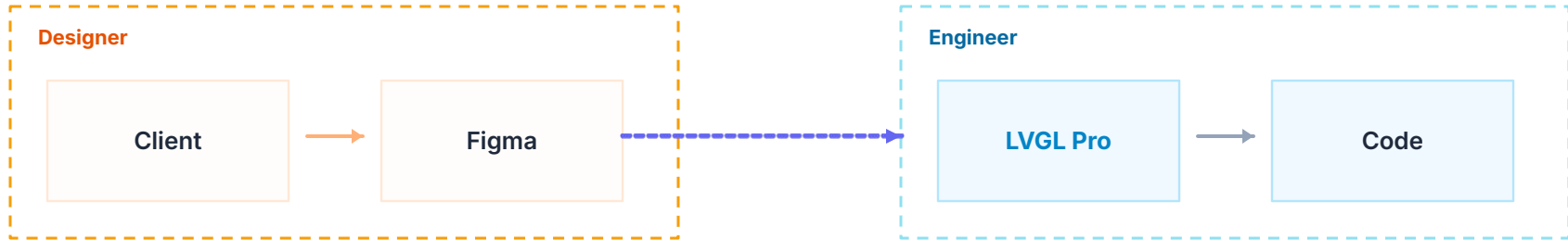
Modern Workflow (**With LVGL Pro** Editor)



Single-Role Workflow (Engineer handles all steps)



Collaborative Workflow (Role-Based Separation of Tasks)



UI Layers

XML Elements

Components, Widgets, Timelines, Screens

Observer & Subjects

Event Handlers (Clicked, Checked, etc.)

Architecture

Design Patterns (MVC, Repository, etc.)

Applications (Core)

BSP and/or LVGL Configuration

Drivers & Subsystems

Hardware Abstraction and Low-level Drivers

Model View Viewmodel (MVVM)

- 01 View**
Binding (styles, flags, state, and text) and Subjects

- 02 View Model**
Observers and notifications (Event callbacks)

- 03 Model**
Bridge between UI side and Application side using public API setters and getters

Model View Controller (MVC)

- 01 Model**
Manages the current application state and updates subjects. In other words, it is C code written to handle the low-level side.

- 02 View**
Touch and key navigation and events with focus management.

- 03 Controller**
Screenshot-based regression testing, so UI changes never slip through unnoticed.

Repository

01 Repository

Raw data from low-level systems (file system, sensors, network status, etc.)

02 Controller / Model

Sends and receives commands from the repository, handling logical decisions (turn on/off, trigger actions, etc.)

03 Subjects

Handles user interactions on the View and dispatches them to the Controller via an observer (value changed, clicked, checked, etc.)

04 View

Contains event handlers and writes directly to the subject side (int, float, string, etc.)

Tools and Channel



Forum

A dedicated place to **share** projects, suggest improvement.



Blog

Tutorials, releases, case studies, and community storie.



Docs

Guides, references, and APIs and topics about Open, Pro, and Safe.



Figma Plugin

Build and automate components, widgets, and screens with just a few clicks.

Electrical Vehicle Demo

Thank you!

Questions? Suggestions?

lvgl.io • lvgl@lvgl.io • github.com/lvgl/lvgl

Contact

[halysson-carvalho](https://github.com/halyssoncarvalho) • github.com/halyssonjr

OBRIGADO!



mouser.com



www.embarcados.com.br



linkedin.com/embarcados



[@portalembarcados](https://www.instagram.com/portalembarcados)



[youtube/Embarcados TV](https://youtube.com/EmbarcadosTV)