LVGL on Oniro

Philippe.Coval +
Gábor.Kiss-Vámosi
#FOSDEM2022
Agenda

- Introduction
- LVGL
- Oniro project
- Integration showcase
Introduction
Introduction

- Philippe Coval is involved into Oniro Project
  - Astrolabe consultant based in France
  - for Huawei Open Source Tech Center
Introduction

- Philippe Coval is involved into Oniro Project
  - Astrolabe consultant based in France
  - for Huawei Open Source Tech Center
- Gábor Kiss-Vámosi is the founder of LVGL
  - CEO of LVGL LLC
  - Developing LVGL and providing Services
What is LVGL?
What is LVGL?

- Light and Versatile Graphics Library
  - Open source (MIT license)
  - Platform independent, and scalable
What is LVGL?

- Light and Versatile Graphics Library
  - Open source (MIT license)
  - Platform independent, and scalable
- First published in 2016 on GitHub
  - Now has 8.3k stars and 200+ contributors
What is LVGL?

- Light and Versatile Graphics Library
  - Open source (MIT license)
  - Platform independent, and scalable
- First published in 2016 on GitHub
  - Now has 8.3k stars and 200+ contributors
- Integrated into many OS and HW solutions:
  - NXP, ESP, NuttX, Zephyr, Rt-Thread, Adafruit, and many more
LVGL UI Demo
LVGL Features
LVGL Features

- Many built in Widgets:
  - buttons, charts, lists, sliders, images, etc.
LVGL Features

• Many built in **Widgets:**
  - buttons, charts, lists, sliders, images, etc.

• Advanced graphics with animations
  - anti-aliasing, opacity, smooth scrolling
LVGL Features

- Many built in **Widgets**:
  - buttons, charts, lists, sliders, images, etc.
- Advanced graphics with animations
  - anti-aliasing, opacity, smooth scrolling
- Fully customizable with **CSS-like styles**
LVGL Features

- Many built in Widgets:
  - buttons, charts, lists, sliders, images, etc.
- Advanced graphics with animations
  - anti-aliasing, opacity, smooth scrolling
- Fully customizable with CSS-like styles
- Supports various input devices:
  - touchpad, mouse, keyboard, encoder, etc.
LVGL Features

- Many built in Widgets:
  - buttons, charts, lists, sliders, images, etc.
- Advanced graphics with animations
  - anti-aliasing, opacity, smooth scrolling
- Fully customizable with CSS-like styles
- Supports various input devices:
  - touchpad, mouse, keyboard, encoder, etc
- Multi-language support with UTF-8 encoding
LVGL Features

- Many built in **Widgets**:
  - buttons, charts, lists, sliders, images, etc.
- Advanced graphics with animations
  - anti-aliasing, opacity, smooth scrolling
- Fully customizable with **CSS-like styles**
- Supports various input devices:
  - touchpad, mouse, keyboard, encoder, etc
- Multi-language support with UTF-8 encoding
- Binding to **MicroPython**
Scalable and Portable
Scalable and Portable

- Written in C for maximal compatibility
  - C++ compatible
Scalable and Portable

- Written in C for maximal compatibility
  - C++ compatible
- Hardware independent:
  - use with any MCU and display
Scalable and Portable

- Written in C for maximal compatibility
  - C++ compatible
- Hardware independent:
  - use with any MCU and display
- OS, external memory, GPU are optional
Scalable and Portable

- Written in C for maximal compatibility
  - C++ compatible
- Hardware independent:
  - use with any MCU and display
- OS, external memory, GPU are optional
- Scalable:
  - from small MCUs (>64 kB Flash, 16 kB RAM)
  - to desktop PCs with 4k displays
Porting LVGL
Porting LVGL

- Only 2 functions are needed to port LVGL:
Porting LVGL

• Only 2 functions are needed to port LVGL:
  1. A **Render ready callback**
     - to copy the rendered image to the screen
Porting LVGL

- Only 2 functions are needed to port LVGL:
  1. A **Render ready callback**
     - to copy the rendered image to the screen
  2. An **Input read callback**
     - to read the touchpad or other device
Porting LVGL

- Only 2 functions are needed to port LVGL:
  1. A **Render ready callback**
     - to copy the rendered image to the screen
  2. An **Input read callback**
     - to read the touchpad or other device
- **Flexible architecture:**
  - you can add GPU support in the pipeline
  - hook any drawing drawing operations
Design with SquareLine
($)$
The Oniro Project

- An independent Eclipse Foundation project
The Oniro Project

- An independent Eclipse Foundation project
- An open source distributed operating system
  - For diverse IoT devices (big and small)
  - Cross kernel, focus on interoperability
The Oniro Project

- An independent Eclipse Foundation project
- An open source distributed operating system
  - For diverse IoT devices (big and small)
  - Cross kernel, focus on interoperability
- Defragment development for embedded sys:
  - Avoid technology silos
  - Unified tooling, common policies,
UI Requirements
UI Requirements

• Oniro can support FLOSS toolkits:
  ▪ Qt, GTK+, EFL... or Web frameworks
UI Requirements

• Oniro can support FLOSS toolkits:
  ▪ Qt, GTK+, EFL... or Web frameworks
• LVGL works on many flavors of Oniro devices
  ▪ A common denominator for CPU/MCU
    ○ For cross kernel Oniro apps
UI Requirements

• Oniro can support FLOSS toolkits:
  ▪ Qt, GTK+, EFL... or Web frameworks
• LVGL works on many flavors of Oniro devices
  ▪ A common denominator for CPU/MCU
    ○ For cross kernel Oniro apps
• Challenge: Prototype app on CPU (Linux)
  ▪ rebase on MCU (Zephyr or Other)
Oniro Linux flavour
Oniro Linux flavour

• Reference "Vending machine" blueprint
Oniro Linux flavour

- Reference "Vending machine" blueprint
- Using lvgl v8 with wayland driver
  - build using bitbake recipe
    - Upstreamed (meta-openembedded)
Oniro Linux flavour

- Reference "Vending machine" blueprint
- Using lvgl v8 with wayland driver
  - build using bitbake recipe
    - Upstreamed (meta-openembedded)
- 2 apps: UI + controller (websockets)
Oniro Linux flavour

- Reference "Vending machine" blueprint
- Using lvgl v8 with wayland driver
  - build using bitbake recipe
    - Upstreamed (meta-openembedded)
- 2 apps: UI + controller (websockets)
- Shipped into customized distro with variables:
  - Screen size, allocator, fonts
  - weston config for kiosk application
Oniro's Vending machine
Oniro Zephyr flavour

- Oniro blueprint: "Keypad device"
  - Custom zephyr-keypad app (WIP)
    - built using bitbake (meta-zephyr layer)
    - shipping zephyr fork of LVGL (v7)
    - work out of the box on nRF52840
Oniro Zephyr flavour

- **Oniro blueprint: "Keypad device"
  - Custom *zephyr-keypad* app (WIP)
    - built using bitbake (*meta-zephyr* layer)
    - shipping *zephyr* fork of LVGL (v7)
    - work out of the box on nRF52840
- **Oniro's plan: *meta-zephyr* (Goofy)
  - Align to v8 : *zephyr*'s lvgl (Thx @brgl)
  - KConfig to customize (thx @pidge)
Oniro Zephyr flavour

- Oniro blueprint: "Keypad device"
  - Custom **zephyr-keypad** app (WIP)
    - built using bitbake (meta-zephyr layer)
    - shipping zephyr fork of LVGL (v7)
    - work out of the box on nRF52840
- Oniro's plan: **meta-zephyr** (Goofy)
  - Align to v8 : zephyr's lvgl (Thx @brgl)
  - KConfig to customize (thx @pidge)
- Challenge: Cross kernel **dialog-lvgl** app
Summary
Summary

- LVGL is portable library for MCU
Summary

- LVGL is portable library for MCU
- Also CPU using drivers: Wayland, SDL
Summary

- LVGL is portable library for MCU
- Also CPU using drivers: Wayland, SDL
- Oniro is crosskernel OS
  - with "blueprint" projects using LVGL
    - Linux: Vending machine
    - Zephyr: Keypad
Summary

• LVGL is portable library for MCU
• Also CPU using drivers: Wayland, SDL
• Oniro is crosskernel OS
  ■ with "blueprint" projects using LVGL
    ○ Linux: Vending machine
    ○ Zephyr: Keypad
• Visit FOSDEM Oniro's stand for more!
Resources and more:

- https://lvgl.io/
  - https://github.com/lvgl/lvgl
- https://oniroproject.org/
  - https://docs.oniroproject.org/
  - https://booting.oniroproject.org/
- https://eclipse.org/
- https://yoctoproject.org/
- https://zephyrproject.org/
Extras?

- Fosdem 2021
- EclipseCon 2021
- SFSCON2021
Howto: Vending machine
Welcome to Oniro

Video Playback