Wayland Desktop Environment

MA35D1/MA35H0

江天文

2024/03/02

Joy of innovation
nuvoTon

Content

Introduction to Wayland

https://wayland.freedesktop.org

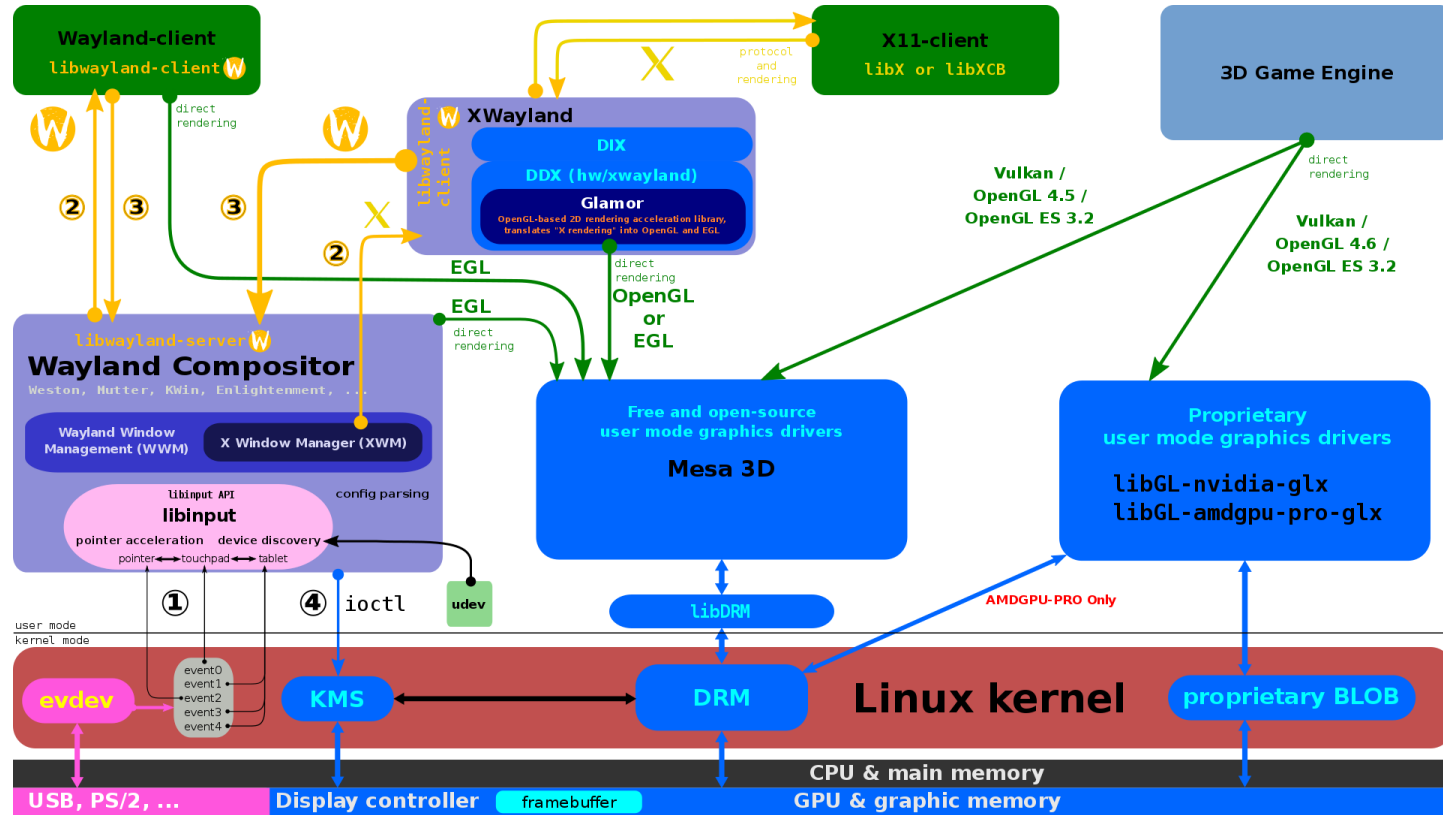# Wayland

Wayland is a communication protocol that specifies the communication between a display server and its client, as well as a C library implementation of that protocol. A display server using the Wayland protocol is called a Wayland compositor, because it additionally performs the task of a compositing window manager. (Wikipedia)

Wayland architecture

Weston

Weston is the reference implementation of a Wayland compositor. When running on Linux, handling of the input hardware relies on evdev. Out of the box, Weston provides a very basic desktop, or a full-featured environment for non-desktop uses such as automotive, embedded, in-flight, industrial, kiosks, set-top boxes and TVs.

It contains a plug-in system of "shell" for common desktop features like docks and panels.

a plug-in Maynard for Weston

Extension protocols to the core protocol

■ **XDG-Shell protocol**

The traditional way to manipulate surface (maximize, minimize, fullscreen) is through Wayland client. XDG shell, on the contrary, is supposed to be provided by the compositor. The XDG (cross-desktop group) shell is a standard protocol extension for Wayland which describes the semantics for application windows.

■ **IVI-Shell protocol**

IVI-Shell is an extension to the Wayland core protocol, targeting in-vehicle infotainment (IVI) devices.

LVGL with Wayland

https://lvgl.io
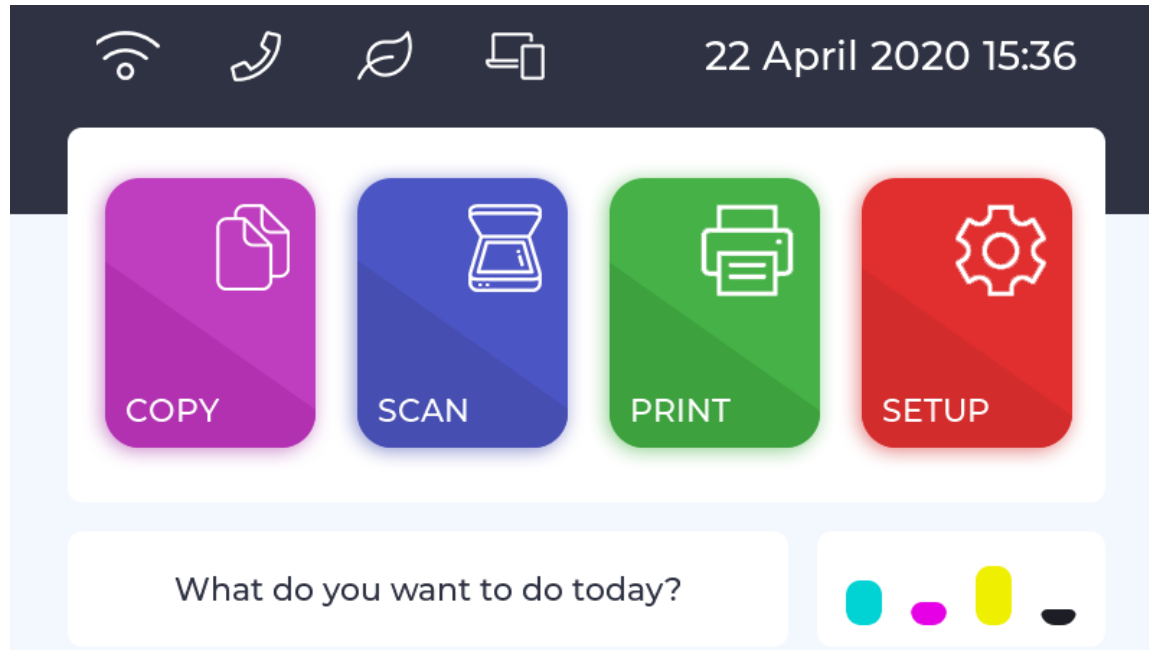
Getting started with LVGL on Wayland

- **Start from scratch by cloning the official Buildroot repository**
  $ git clone https://github.com/OpenNuvoton/MA35D1_Buildroot.git

- **Change directory to *the root of Buildroot* (${BR2_DIR}), and clone repository *ma35d1-portal* on branch *weston-10.0.5*.**
  $ cd ${BR2_DIR}
  $ git clone -b weston-10.0.5 https://github.com/symfund/ma35d1-portal.git

- **Delete original packages, and replace with corresponding packages from *ma35d1-portal/package* on branch *weston-10.0.5*.**
  $ cd ${BR2_DIR}/package; rm -Rf cairo eudev libdrm libinput libpng libxkbcommon mesa3d-headers mesa3d pixman wayland-protocols wayland-utils wayland weston; cp -Rf ../ma35d1-portal/package/wayland wayland; …

Configuring Buildroot to import packages

- **Clone *buildroot-external* into the root of Buildroot**
  $ git clone https://github.com/symfund/buildroot-external.git
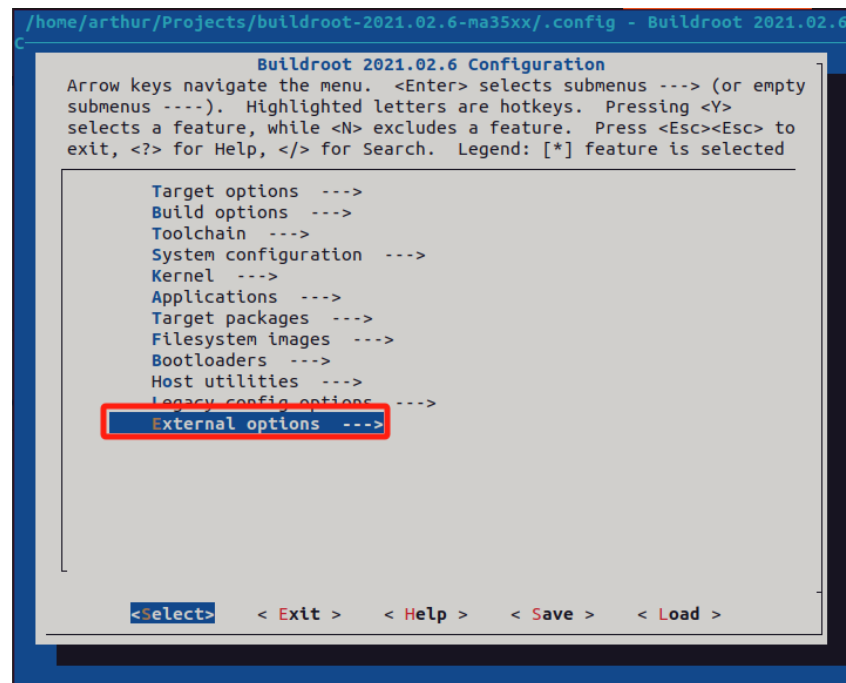
- **Load a specific board default configuration**
  $ make numaker_som_ma35d16a81_defconfig

- **Configure Buildroot**
  $ make BR2_EXTERNAL=buildroot-external menuconfig

Configuring Buildroot to manage /dev

- **/dev management** using *devtmpfs + eudev*

  →System configuration
  /dev management
  ( ) Static using device table
  ( ) Dynamic using devtmpfs only
  ( ) Dynamic using devtmpfs + mdev
  (X) Dynamic using devtmpfs + eudev

11

Configuring Buildroot to enable mesa3d

- **Select *mesa3d***
  - **Gallium swrast driver**
  - **OpenGL EGL**
  - **OpenGL ES**

Configuring Buildroot to select free fonts

- **Free fonts**
  - **mono fonts**
  - **sans fonts**
  - **serif fonts**

Configuring Buildroot to enable weston

- **Select *weston***

  - **default compositor: fbdev**

  - **desktop shell**

  - **fullscreen shell**
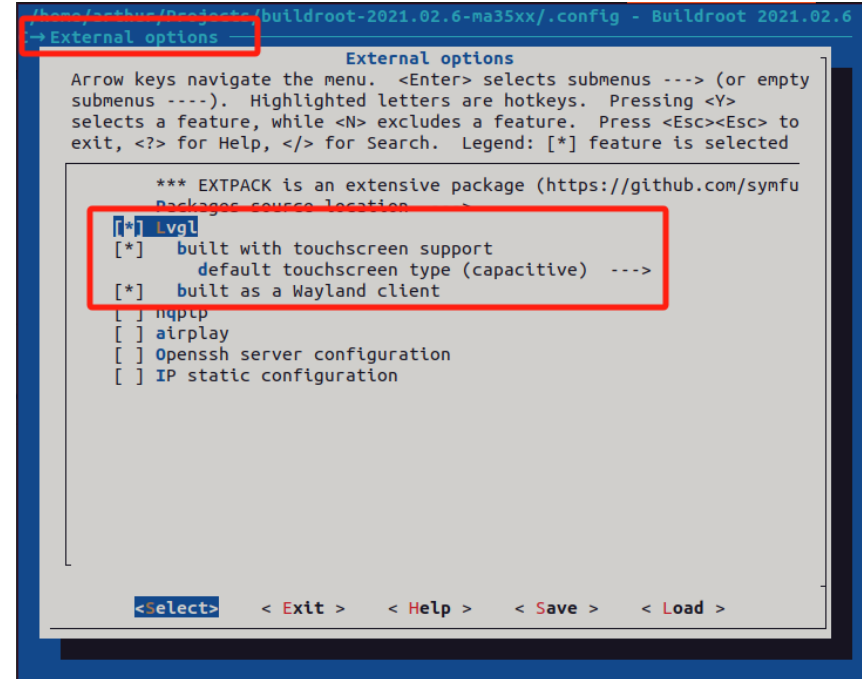
Configuring Buildroot to enable LVGL

- **Select *lvgl***

  Choose the right *default touchscreen type*, capacitive or resistive.

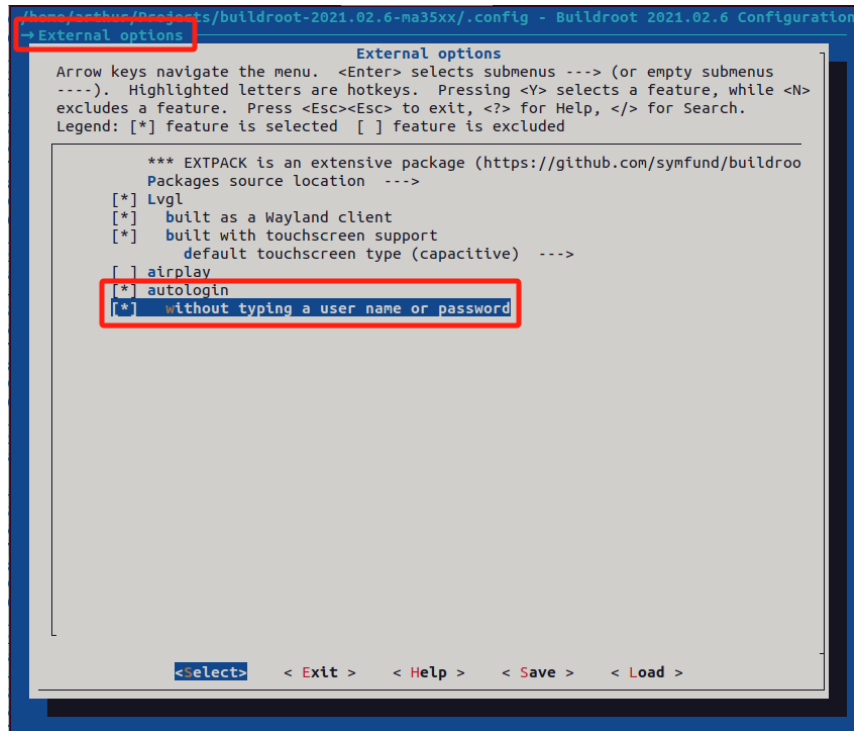 With Wayland enabled, LVGL should always be *built as a Wayland client*.
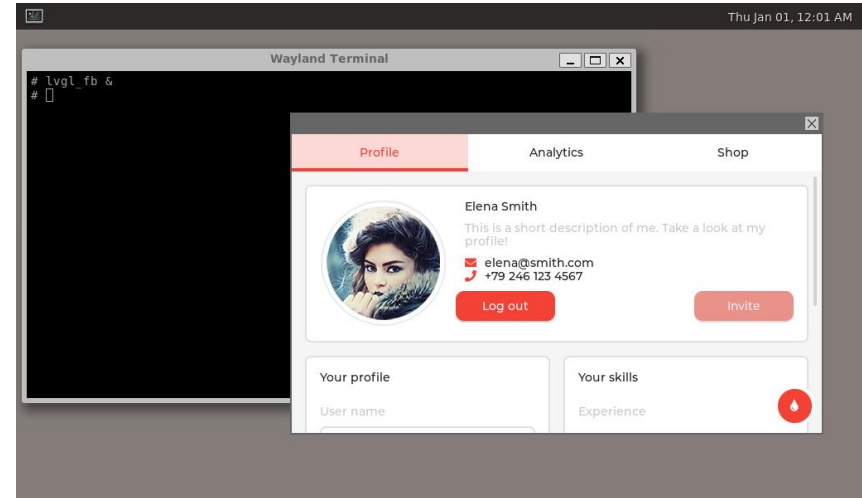
Auto login without typing user name

- **Auto-login logs into Wayland/Weston without typing a user name or password**

Runing LVGL on Wayland

- **Touch the terminal icon in the title bar to launch the weston-terminal.**

- **On the command line, type 'lvgl_fb &' to run LVGL application.**
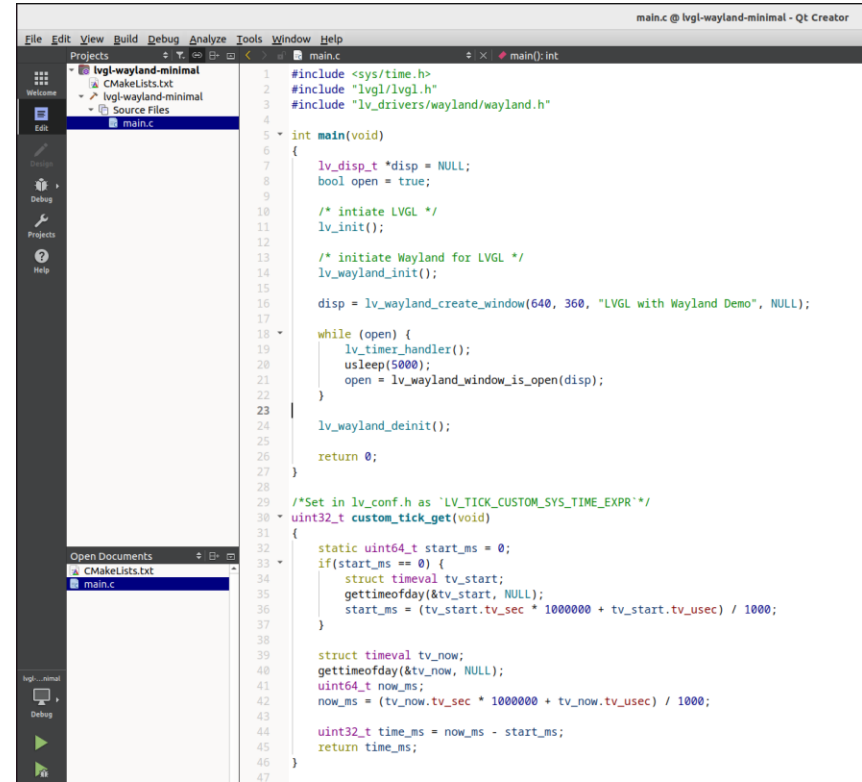
The skeleton minimal LVGL with Wayland

- **The skeleton example**

  ma35d1-portal/examples/wayland/lvgl/lvgl-wayland-minimal

- **Modify *CMakeLists.txt*, set variable *BR2_DIR* to the actual path of Buildroot. If Qt Creator Kits use SDK toolchain, just set *CMAKE_SYSROOT* to the path of SYSROOT.**

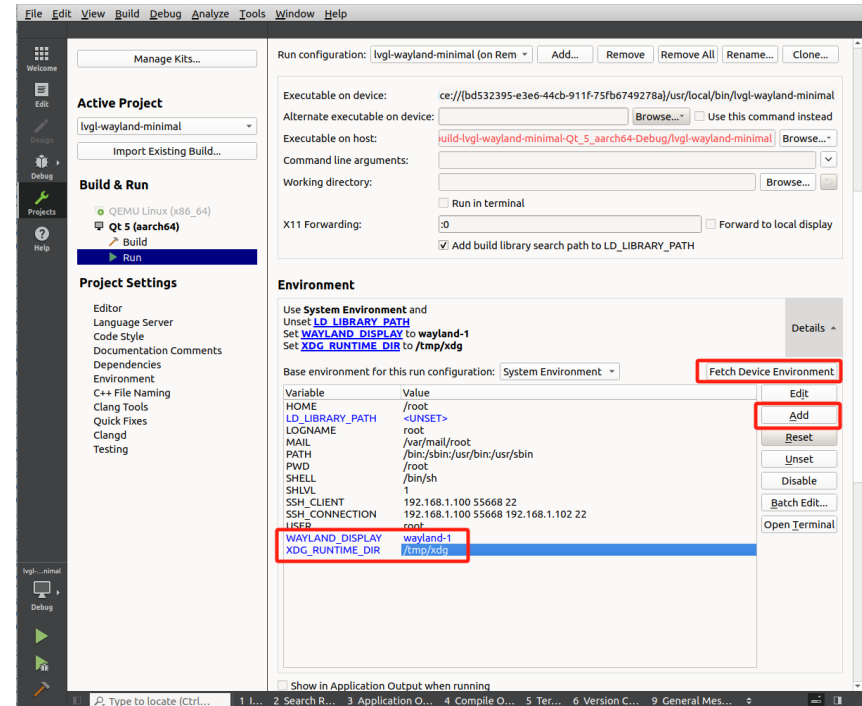 set(BR2_DIR "/path/to/buildroot")

  set(CMAKE_SYSROOT "/path/to/sysroot")

Remote debugging with LVGL on Wayland
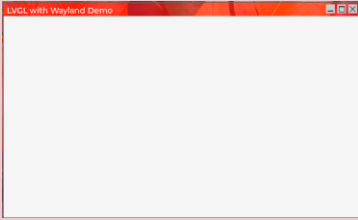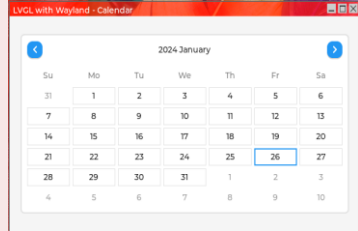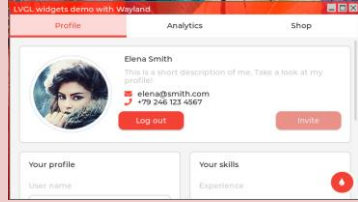
- **Fetch device environment**

- **Add two variables**

  - **XDG_RUNTIME_DIR** (/tmp/xdg)

  - **WAYLAND_DISPLAY** (wayland-1)

LVGL minimal with Wayland
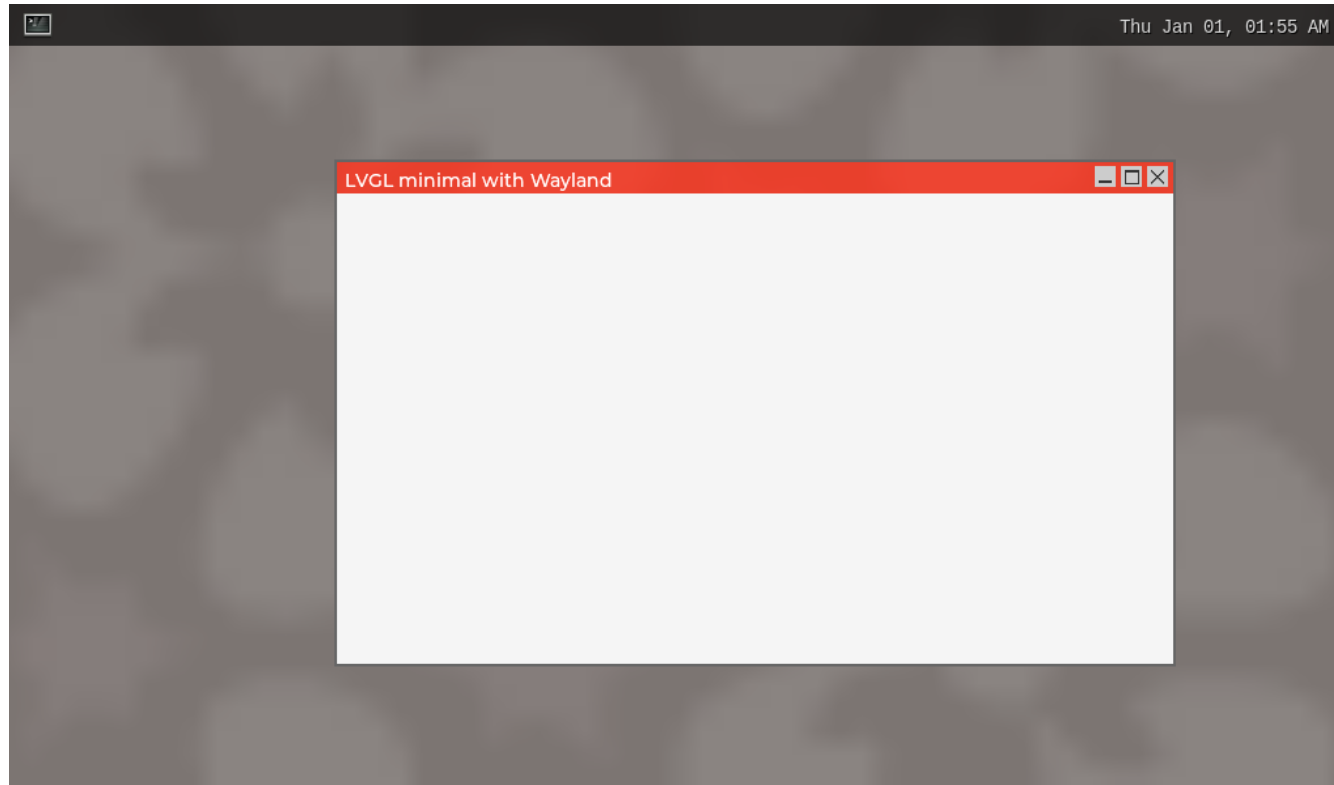
LVGL examples with Wayland

Apps	Location (https://github.com/symfund/ma35d1-portal)	Thumbnails
Minimal	examples/wayland/lvgl/lvgl-wayland-minimal	
Calendar	examples/wayland/lvgl/lvgl-wayland-calendar	
Widgets	examples/wayland/lvgl-wayland-widgets-demo	

Qt5 with Wayland

https://www.qt.io

Configuring Buildroot to enable Qt5

- **Select Qt5**

  Component qt5wayland requires *OpenGL support*. OpenGL API uses *OpenGL ES 2.0+*.

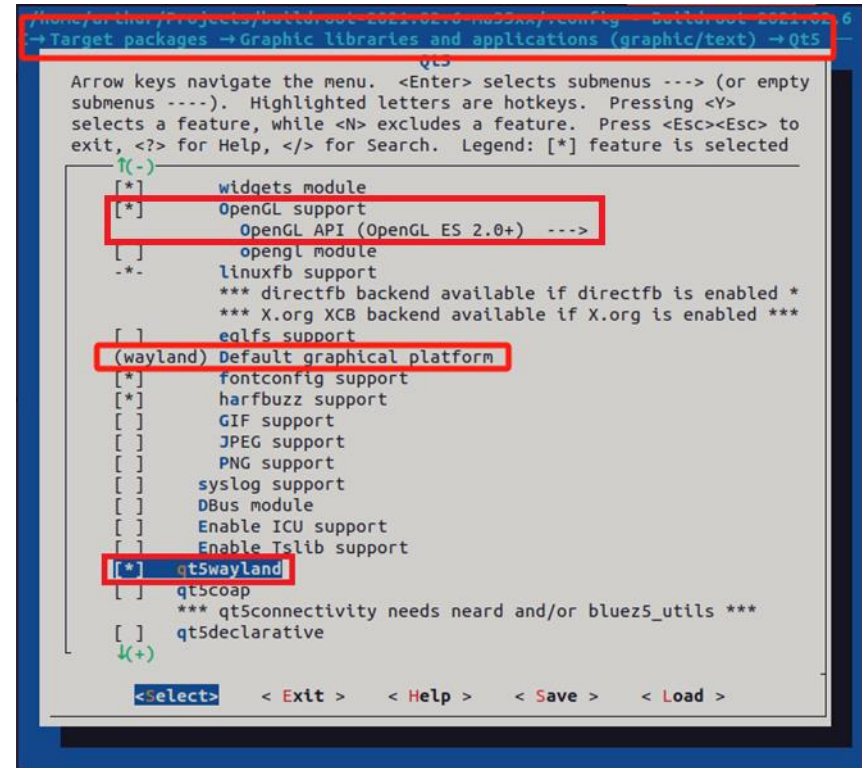 The *default graphical platform* is set to *wayland* or *wayland-egl*.

  With wayland enabled, *qt5wayland* component should always be selected.
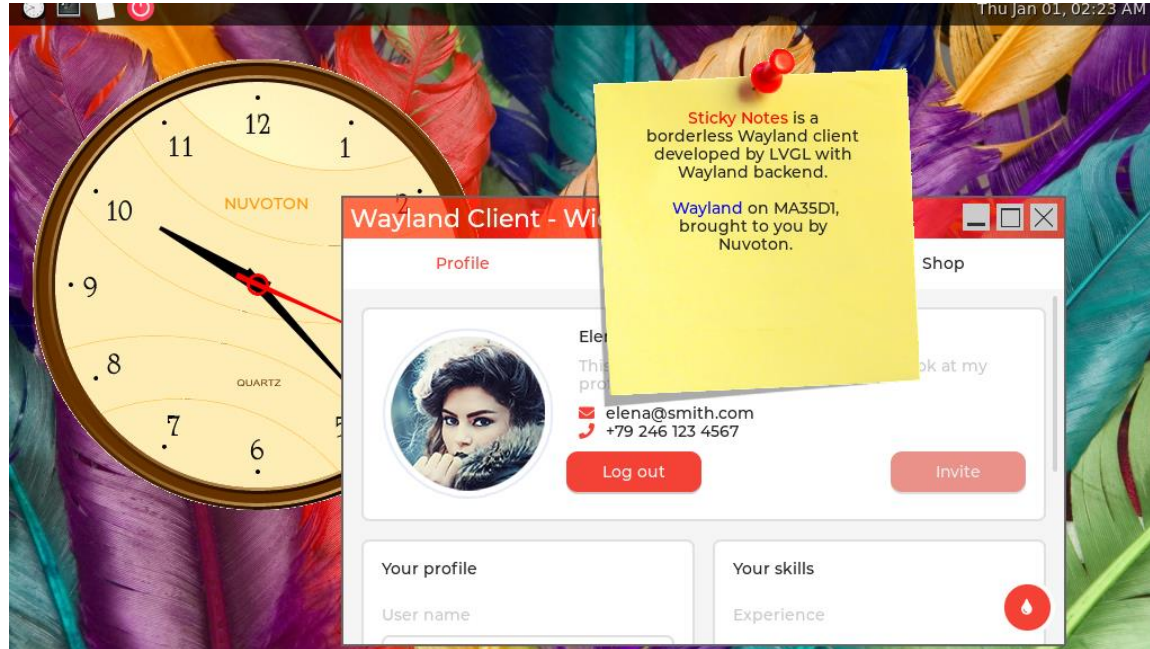
Qt5 Icons example on Wayland

Customizing Weston

Changing wallpapers, adding launch icons, setting idle lock time

Changing wallpaper

- **Prepare a wallpaper on target location /usr/share/weston/wallpaper.jpg**

- **Edit /etc/xdg/weston/weston.ini, under section 'shell', modify the entry 'background-image'**

  [shell]
  background-image=/usr/share/weston/wallpaper.jpg
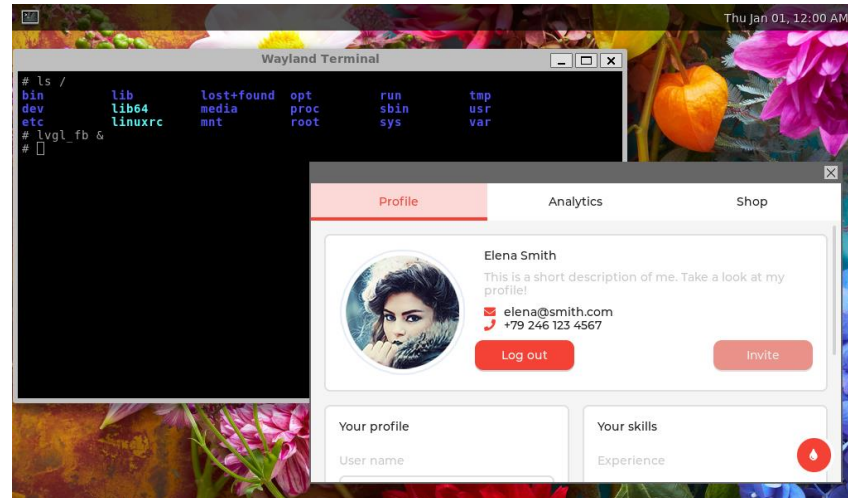
Modifying task panel color

- Edit **/etc/xdg/weston/weston.ini**, under section **'shell'**, modify the entry **'panel-color'**
  [shell]
  panel-color=0xffff0000

- Add translucent color
  [shell]
  panel-color=0x12ff0000
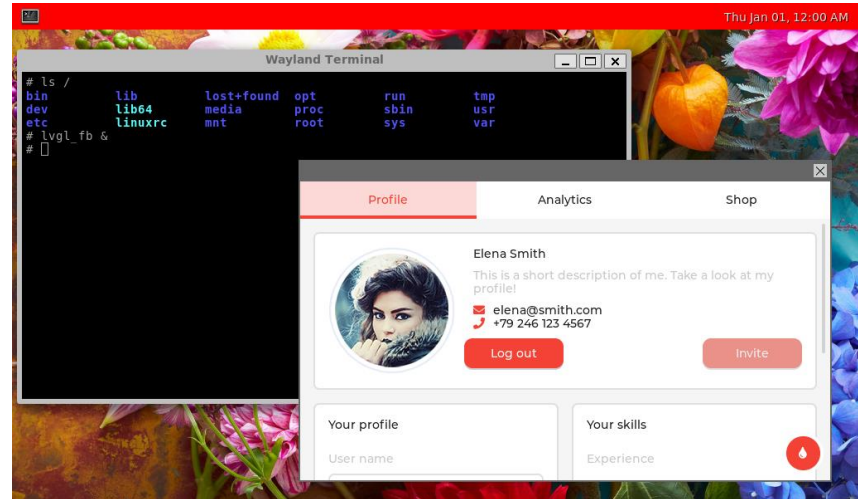
Adding launch icons in task panel

- Edit **/etc/xdg/weston/weston.ini**, add section **'launcher'**, under section 'launcher', append entries **'icon'** and **'path'**

  [launcher]
  icon=/usr/share/weston/wayland24.png
  path=/usr/bin/lvgl_fb

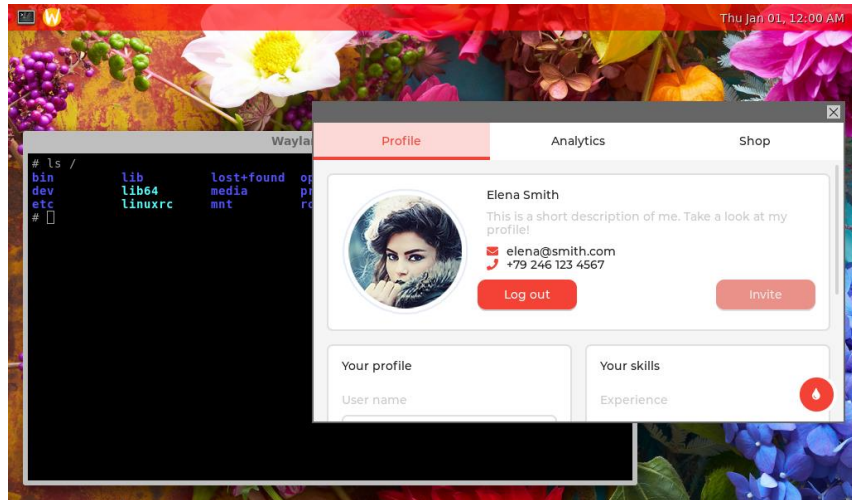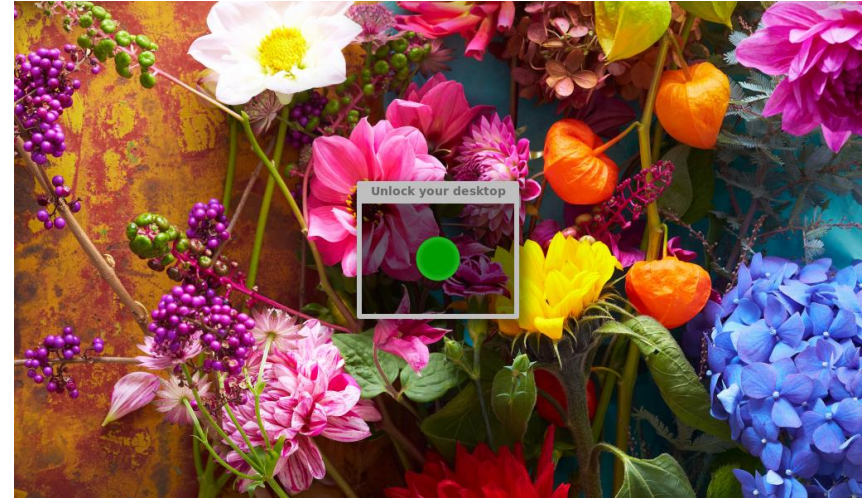# Setting idle lock time

- **Edit /etc/xdg/weston/weston.ini, under section 'core', modify the entry 'idle-time' (seconds)**

  [core]
  idle-time=300

Irregular non-rectangle shaped window

- **Some irregular non-rectangle shaped window applications like Analog Clock require LVGL to run with window decorations (title, border) disabled. This goal can be achieved by setting environment variable LV_WAYLAND_DISABLE_WINDOWDECORATION to "1" before launching the shaped window.**

  $ LV_WAYLAND_DISABLE_WINDOWDECORATION=1 lvgl-wayland-analog-clock &