ivi-homescreen

Toyota Connected North America - UI/UX Team

Joel Winarske

Flutter

- Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, desktop, and embedded devices from a single codebase.
- Dart is a programming language designed for client development, such as for the web and mobile apps.
- Skia is an open source 2D graphics library which provides common APIs that work across a variety of hardware and software platforms.
- Flutter is the combination of Dart + Skia.

ivi-homescreen

- Flutter embedder specifically for Wayland
- Strongly Typed (C++)
- Lightweight
  - Clang 11 Release Stripped = 151k
  - GCC 9.3 Release Stripped = 168k
- Runs on Linux Desktop and Yocto Linux
  - Ubuntu 18+
  - Fedora 33+
  - Yocto Dunfell+
- Platform Channels enabled/disabled via CMake
- OpenGL Texture Framework

Developer Workflows

- Execute/Debug Application on Ubuntu/Fedora Desktop
- Execute/Debug Flutter Application on Target

Execute Application on Desktop

- Install ivi-homescreen and flutter-engine packages
- Build your flutter application using flutter build bundle
  - cd ~/development/my_flutter_app
  - flutter channel beta
  - flutter upgrade
  - flutter config --enable-linux-desktop
  - flutter create .
  - flutter build bundle
- Create application symlink in /usr/local/homescreen
  - cd /usr/local/share/homescreen && sudo rm -rf bundle/
  - sudo ln -sf ~/development/my_flutter_app/build/ bundle
- Run Flutter Application
  - /usr/local/bin/homescreen

Debug Application on Desktop

- Follow steps under Execute Application on Desktop with following exception
  - Install flutter-engine “debug” package
- After Flutter embedder is running attach to process
  - flutter attach --debug-port 41795 --host-vmservice-port 41795
- Hot Reload, Restart, etc. is available via console only

Working with a Yocto OS Image

- Create Yocto image using meta-flutter layer including ivi-homescreen and flutter-engine.
- Install Ahead Of Time (AOT) Flutter Application image on target. Either manually or via recipe as part of image build. From command window run /usr/bin/homescreen
- Notes:
  - Flutter applications built by default are machine agnostic, Ahead Of Time (AOT) compiled Flutter images are machine specific.
  - For application Yocto recipe pattern see: https://github.com/jwinarske/meta-flutter/blob/dunfell/recipes-graphics/flutter-apps/flutter-gallery_git.bb

AGL Build Example

- Creates Emulation image that includes ivi-homescreen and libflutter_engine.so
  - export MACHINE= qemux86-64
  - repo init -b lamprey -u https://gerrit.automotivelinux.org/gerrit/AGL/AGL-repo --no-clone-bundle
  - repo sync --no-clone-bundle -j `grep -c ^processor /proc/cpuinfo` &
  - git clone -b dunfell https://github.com/kraj/meta-clang.git &
  - git clone https://github.com/jwinarske/meta-flutter &
  - wait
  - source meta-agl/scripts/aglsetup.sh -m $MACHINE agl-devel agl-basesystem
  - echo "IMAGE_INSTALL_append = \" ivi-homescreen\"" >> conf/local.conf
  - bitbake-layers add-layer ../meta-clang ../meta-flutter
  - bitbake agl-image-weston

Create Ahead Of Time (AOT) image

- flutter build bundle
- dart ${ENGINE_SDK}/frontend_server.dart.snapshot --aot --tfa --target=flutter --sdk-root ${ENGINE_SDK} --output-dill app.dill lib/main.dart
- ${ENGINE_SDK}/clang_x64/gen_snapshot --deterministic --snapshot_kind=app-aot-elf --elf=libapp.so --strip app.dill
- Notes:
  - Requires flutter-engine “Release” variant to run
  - Desktop ENGINE_SDK=/usr/local/share/flutter/sdk
  - Target ENGINE_SDK=/usr/share/flutter/sdk
  - See integration example here: https://github.com/jwinarske/meta-flutter/blob/dunfell/recipes-graphics/flutter-apps/flutter-gallery_git.bb

Important Notes

- In order to run a Flutter application, the Flutter SDK Dart VM version must match the Flutter Engine Dart VM version.
    - You can check the Flutter Engine Dart VM version via:
        - Desktop
            - cat /usr/local/share/flutter/sdk/dart.version
        - Target
            - cat /usr/share/flutter/sdk/dart.version
- A flutter channel will roll or change commit hash periodically. The channels are master, dev, beta, stable. The rate of change is greatest with the master channel, where as the beta channel rolls generally once a week.

x86_64 binary packages

- ivi-homescreen
  - https://github.com/toyota-connected/ivi-homescreen/actions/runs/1251358078
- flutter-engine
  - https://github.com/jwinarske/flutter_embedded/actions/runs/1259970904
- Deployment to JArtifactory based Debian/RPM/Opkg repository, including versioning and dependent packages planned. Possibly an alternative that is hosted on AGL site?

Repositories

- https://github.com/toyota-connected/ivi-homescreen
  - Linux Flutter Embedder
- https://github.com/jwinarske/flutter_embedded
  - X86_64 Debian, RPM, tar.gz package of libflutter_engine.so
  - Runtime – debug, profile, release
- https://github.com/jwinarske/meta-flutter
  - Yocto layer for embedded Flutter
  - includes recipes for flutter-engine and ivi-homescreen
- https://github.com/jwinarske/flutter-channel-watch
  - Checks flutter channel every 15 minutes for a change in commit
  - Used to trigger flutter-engine builds

Planned/Active Work

- Open Source AGL integration
  - Full featured homescreen
- v2 ivi-homescreen
  - Vulkan Based
  - 3D Rendering engine integration
  - Flutter Engine renders to RenderTarget of type vkImage