Minimize AGL For Low Profile

July 18 2019
Automotive Linux Submit, Tokyo

Zhou Mingying
Nanjing Fujitsu Nanda Software Tech. Co., Ltd.(FNST)
Self Introduction

Zhou Mingying
Project Manager
Senior engineer
Linux Driver/Service/ and Application
FNST Automotive Business

AI and Etc. Advanced Tech.

In-Vehicle Embedded

In Vehicle Device
Drive Record
Digital・Tachograph
DCM/TCU/T-BOX

Multimedia
Multimedia Service
Navigation
Infotainment Device

Automotive Electronics

Engine

Power Train Control
Engine cooling System

Vehicle Info

Power Train Control

Mobility as a Service

Cloud

Error Analysis
Runtime Manage System
MaaS Platform

Control ECU

Engine ECU
Transmission ECU
fastECU

Intelligent Vehicle

Copyright 2019 NANJING FUJITSU NANDA SOFTWARE TECHNOLOGY CO., LTD.
FNST AGL work

AGL Community

Apps

- HVAC
- Radio
- Telephone
- Navigation
- HVAC
- BlueTooth
- Weather
- SDL
- WIFI

FW/Services

- Native HMI
- APP FW
- Web HMI
- Platform
- Automotive
- Other

OS

- Kernel
- Boot
- BSP

CIAT Tool

- LAVA
- Fuego
- Kselftest
- AGL-JTA

FNST relevant work
Agenda

- Why minimize AGL
- How to minimize AGL
- The effect of AGL minimization
- What to do next
Why minimize AGL

- Low-end SoC
- Profile Architecture
Low-end SoC

- The status of AGL now (stable AGL7.0.3)
  - High-end is main now, such as R-Car H3,M3
  - Action on low-end SoC is slow because of a lot of Apps and Services

- What we want to do
  - low-end, such as E3,D3 for low-profile in emerging countries
  - Less applications, just vehicle information and smart phone connection, such as
    - `agl-service-can-low-level`
    - `agl-service-hvac`
    - `SDL(SmartDeviceLink)`
How to minimize AGL

- Add a new image option
- Remove apps and add services
- Replace compositor
- Simulate low-end SoC with M3
Add a new image option

- Add “agl-demo-minimal”
- add bb file to meta-agl-cluster-demo

```bash
$ ls meta-agl-cluster-demo/recipes-platform/images/
agl-demo-minimal-crosssdk.bb  agl-demo-minimal.bb  agl-demo-minimal.inc
```

- For agl-demo-minimal-crosssdk option
- For agl-demo-minimal option
- Include file

Though add the new option, no effect to other existing images
Remove apps and add services

- Add bb file for agl-demo-minimal image
  - recipes-platform/packagegroups/packagegroup-agl-demo-minimal.bb
    - No effect to existing images
- Define necessary components
  - Other components will be excluded
- Qt Lite
  - When remove apps, the dependency Qt components will be excluded
  - Remove charts, virtual board and xml components, etc.
- Remove unused language packages
  - Chinese
  - Japanese
- Patch path (Under review):
What’s Wayland compositor

- Wayland is a display server and compositing protocol
- Wayland compositor is the implementation of the protocol

Note: Picture is from https://wayland.freedesktop.org/docs/html/ch01.html#sect-Compositing-manager-display-server
Replace compositor(2)

- Replace Weston with QtWayland compositor

- 3 processes(①②③) => 1 process(④)
- Customized compositor is more flexible

Customized compositor. But protocol ivi-controller and ivi-wm aren’t supported.
Replace compositor(3)

Action items

- Build plugin
- Create QtWayland compositor sample
  - Cluster Demo (Cluster-Dashboard)
- Add configure file kms.conf
- Create QtWayland compositor service
Build plugin

- Compile qtbase with eglfs, gbm and kms (AGL7.0.3)
  - Step 1: modify qtbase_%.bbappend
    ```bash
    PACKAGECONFIG_GL = "${@bb.utils.contains('DISTRO_FEATURES', 'opengl', 'gles2', 'eglfs kms gbm', '', d)}"
    ```
  - Step 2: build
    ```bash
    $source meta-agl/scripts/aglsetup.sh -m m3ulcb -b build agl-cluster-demo
    $bitbake qtbase
    
or
    $source meta-agl/scripts/aglsetup.sh -m m3ulcb -b build agl-cluster-demo
    $bitbake agl-demo-minimal
    ```

- Plugin libqeglfs-kms-integration.so
  - After building, the plugin libqeglfs-kms-integration.so is included by qtbase-plugins-5.9.7+git0+81b29a44d2-r0.aarch64.rpm
  - Install plugin
    - Install the qtbase-plugins-5.9.7+git0+81b29a44d2-r0.aarch64.rpm
    - Rewrite image file agl-demo-minimal
Create QtWayland compositor

Create a new sample compositor Cluster-Dashbord
  - Delete window manager relevant code
  - Create bb file
    - recipes-demo-hmi\cluster-dashboard-sample\cluster-dashboard-sample_git.bb
  - Compile and install

Stop Weston

```
#killall weston
```

- If have removed Weston and window manager, “killall Weston” isn’t needed

Start QtWayland compositor

- Cluster-Dashboard-sample

If create a service, it can be started when the OS start

```
#/var/local/lib/afm/applications/cluster-gauges/1.0-923279f-dirty/bin/cluster-gauges -platform eglfs
```

Before starting QtWayland compositor, should make a kms.conf file
Add configure file kms.conf

kms.conf (example)

```bash
m3ulcb:~# cat /etc/kms.conf
{
    "device": "/dev/dri/card0",
    "outputs": [
        {
            "name": "LVDS1",
            "mode": "off"
        }
    ]
}
```

You can get some sample from the bellow address:

You can define the kms.conf file as you need.
Create QtWayland compositor service

- Add detail to bb file
  - recipes-demo-hmi\cluster\dashboard-sample\cluster\dashboard-sample_git.bb

```bash
inherit systemd
SRC_URI_append = "
   file://0001-Modify-cluster-dashboard-as-a-qtwayland-compositor.patch
   file://0002-Add-kms-conf-file.patch
   file://qtwaylandcompositor.service"

do_install_append() {
    install -d ${D}/usr/bin
    install -m 755 ${WORKDIR}/build/app/cluster-gauges ${D}/usr/bin
    install -d ${D}/etc
    install -m 644 ${S}/kms.conf ${D}/etc
    install -d ${D}${systemd_unitdir}/system
    install -m 644 ${WORKDIR}/qtwaylandcompositor.service ${D}${systemd_unitdir}/system/qtwaylandcompositor.service
}

SYSTEMD_SERVICE_${PN} = "qtwaylandcompositor.service"
```

Add the above code to bb file, the service will be installed at the first start of AGL. And can be started from next OS start.

Add kms.conf file

Remove window manager relevant code

Add compositor service file
Simulate low-end SoC with M3

- We don’t have Low SoC, use M3 to instead of low SOC
- CPU&Memory limit set to simulate Low SoC

- Bootargs modify

  ⇒ editenv bootargs
  edit: console=ttySC0,115200 root=/dev/mmcblk1p1 rootwait ro
  rootfstype=ext4 maxcpus=1 mem=896M

- DTS file modify (example)
The effect of AGL minimization

- Demo screen
- Packages & ROM & RAM
- Performance
Demo Screen

- Same as AGL’s agl-cluster-demo-platform’s screen
### Packages & ROM & RAM

- **Packages & ROM & RAM is smallest on ②SDL+Vehicle**
- **Remove Window Manager and Languages, Qt Lite**

<table>
<thead>
<tr>
<th></th>
<th>① AGL Cluster profile</th>
<th>② SDL+Vehicle info (agl-demo-minimal)</th>
<th>③ AGL IVI FULL profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packages Num.</td>
<td>1328</td>
<td>970</td>
<td>1593</td>
</tr>
<tr>
<td>Component Num.</td>
<td>269</td>
<td>205</td>
<td>397</td>
</tr>
<tr>
<td>ROM (MB)</td>
<td>924</td>
<td>471</td>
<td>1144</td>
</tr>
<tr>
<td>RAM (MB)</td>
<td>348</td>
<td>284</td>
<td>896</td>
</tr>
</tbody>
</table>

Test version AGL7.0.2
Performance

- CPU(%) is smallest on ②SDL+Vehicle
- Remove Weston&Window Manager (① ③ use window Manager)
- Less applications

<table>
<thead>
<tr>
<th>Actions</th>
<th>① AGL Cluster profile</th>
<th>② SDL+Vehicle info agl-demo-minimal</th>
<th>③ AGL IVI FULL profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Graphics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Graphics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Graphics (no homescreen)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Test version AGL7.0.2
- Test after booting(Cluster Graphics), no loaded
- Boot with 2CPU on M3 (simulate E3)
What to do next?

- Help Qt to support other protocols
- Modify applications
Help Qt support other protocol

- Qt supports ivi-application only (2019.7)
- Other protocol

  Weston support the bellow protocols
  - ivi-controller
    - AGL used ivi-controller before 2018.10
  - ivi-wm
    - From 2018.2, AGL start to use ivi-wm protocol

Make Qt Wayland Compositor to support ivi-wm, This will help QtWayland compositor more flexible. The existing applications don’t need to modify.
Modify applications

- Modify application
  - The original applications depend on window manager
  - After removing window manager, the application should be modified
Questions?
Thank you!

zhoumy@cn.fujitsu.com