Introduction to Automotive Grade Linux

AGL Fall AMM 2016

Walt Miner ( @VStarWalt )
Community Manager, AGL, The Linx Foundation
Automotive Grade Linux

Collaborating to build the car of the future through rapid innovation

http://AutomotiveLinux.org
### Git Commits BB and CC

<table>
<thead>
<tr>
<th>Commits</th>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>458</td>
<td>Jose Bollo</td>
<td>IoT.BZH</td>
</tr>
<tr>
<td>341</td>
<td>NuoHan Qiao</td>
<td>Fujitsu Ten</td>
</tr>
<tr>
<td>70</td>
<td>Stephane Desneux</td>
<td>IoT.BZH</td>
</tr>
<tr>
<td>64</td>
<td>Ran Cao</td>
<td>Fujitsu Ten</td>
</tr>
<tr>
<td>59</td>
<td>Manuel Bachmann</td>
<td>IoT.BZH</td>
</tr>
<tr>
<td>58</td>
<td>Jan-Simon Moeller</td>
<td>Linux Foundation</td>
</tr>
<tr>
<td>55</td>
<td>Fulip Ar Foll</td>
<td>IoT.BZH</td>
</tr>
<tr>
<td>35</td>
<td>Yanhua GU</td>
<td>Fujitsu Ten</td>
</tr>
<tr>
<td>34</td>
<td>Christian Gromm</td>
<td>Microchip</td>
</tr>
<tr>
<td>27</td>
<td>Yannick Gicquel</td>
<td>IoT.BZH</td>
</tr>
<tr>
<td>20</td>
<td>Tadao Tanikawa</td>
<td>Panasonic</td>
</tr>
<tr>
<td>15</td>
<td>Leon Anavi</td>
<td>Konsulko</td>
</tr>
<tr>
<td>7</td>
<td>Kotaro Hashimoto</td>
<td>Mitsubishi Electric</td>
</tr>
<tr>
<td>6</td>
<td>Yuta Doi</td>
<td>Witz</td>
</tr>
<tr>
<td>5</td>
<td>Stephen Lawrence</td>
<td>Renesas</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Commits</th>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Andre Magalhaes</td>
<td>Collabora</td>
</tr>
<tr>
<td>4</td>
<td>Phong Tran</td>
<td>Renesas</td>
</tr>
<tr>
<td>3</td>
<td>Anton Gerasimov</td>
<td>Advanced Telematics</td>
</tr>
<tr>
<td>3</td>
<td>Jens Bockage</td>
<td>Mentor</td>
</tr>
<tr>
<td>2</td>
<td>Carlos Alberto Perez</td>
<td>Igalia</td>
</tr>
<tr>
<td>2</td>
<td>Tomoki Sekiyama</td>
<td>Hitachi</td>
</tr>
<tr>
<td>1</td>
<td>Wataru Natsume</td>
<td>ADIT</td>
</tr>
<tr>
<td>1</td>
<td>Philippe Coval</td>
<td>Samsung</td>
</tr>
<tr>
<td>1</td>
<td>Tasuku Suzuki</td>
<td>Qt Company</td>
</tr>
<tr>
<td>1</td>
<td>Damian Hobson-Garcia</td>
<td>Renesas</td>
</tr>
</tbody>
</table>

*Since 15 Jan 2016

1260 Total Commits
18 Companies
AGL Distro “Unified Code Base”

- First Release announced at CES Las Vegas in January
- Unifying the best of AGL, Tizen IVI and GENIVI into a single code base for the entire industry!
- Reduce fragmentation, focus on innovation and new features!
- Yocto/Poky based with AGL specific layers
Thanks for all the fish...

- AGL Releases:
  - AA – Agile Albacore – Jan 2016
  - BB – Brilliant Blowfish – July 2016
  - CC – Charming Chinook – Jan 2017
  - DD – Daring Dab – July 2017
Brilliant Blowfish

- Released July 15, 2016
- Upgrade to Yocto 2.0
- Additional BSPs
- IVI Audio Manager
- IVI Layer Manager
- Automated Test Improvements
Brilliant Blowfish

- Reference BSPs – Fully supported by manufacturer, CI, etc.
  - Renesas R-Car 2 - Porter board – **Full ALS demo**
  - QEMU – demo code available – not shown at ALS
- Community BSP – Best effort by AGL with minimal support
  - NXP – i.MX6 – SABRE – **ALS demo available**
  - NXP - i.MX6x – Wandboard – issues with graphics drivers
  - Intel - Minnowboard Max - demo code available – not shown at ALS
  - TI - Jacinto 6 - Vayu board – **ALS demo available**
  - QCOM – Dragonboard 610-c – no demo available
  - Raspberry PI – no demo available
Reference or Community BSP?

- **Reference board**
  - BSP available as part of AGL Core Distribution
  - BSP maintained by board manufacturer
  - Documentation and Kick-start guide available for downloading and building code and running the AGL demo code.
  - SDK Released and maintained
  - Manufacturer provides at least two boards for AGL Continuous Integration and Automated Test (CIAT) infrastructure
- **Continuous Integration**
  - Daily snapshot builds available from AGL Jenkins
- **Test and QA**
  - Sponsoring company sets up test nodes in Lava
  - Full AGL CIAT test suite is run
  - Test results reported.
  - Expect >90% pass
Reference or Community BSP?

- Community board
  - Hobbyist boards that are not automotive specific
  - Older automotive specific boards that are no longer sponsored / maintained by the manufacturer
  - Best effort by the community
  - AGL will have “featured” community BSP(s) as proposed by the community and designated by the SAT

- See [https://wiki.automotivelinux.org/agl-distro#supported_hardware](https://wiki.automotivelinux.org/agl-distro#supported_hardware) for list of boards
Patch Releases

- Brilliant Blowfish 2.0.1 patch release available
- BB 2.0.2 will be available Week 37
Charming Chinook

• Target December 15, 2017
• Yocto 2.1
• SDK available
• Reference AGL Apps
• AGL Compositor
• AGL Home Screen Reference App in Qt and HTML5
• Device Profiles for Telematics, IC, ADAS
• IP Network Manager with WiFi and LTE
Charming Chinook

• Timeline
  • Yocto 2.1 (Kergoth) merged to master
  • Master open for feature integration
  • Core Distribution Feature Freeze 01 Nov 2016
  • RCs every 2 two weeks after that
  • Release 31 Dec
CODE STRUCTURE
Software Configuration Requirements

Readily determine the required contents of the AGL distribution for product developers
AGL Core Distribution

- Stable Yocto release
- Reference BSPs fully supported by the board manufacturer or chip vendor
- Documentation and tooling for building and deploying reference BSPs
- Tooling to allow selection of optional features in the core build
- Test results provided using AGL Test Framework
- Fully supported with updates for at least 6 months
- Defined by Yocto layer – meta-agl
Provide a mechanism for enabling optional and/or experimental features
AGL Extra Features

- Builds on AGL Core Distribution
- Features are fully tested and supported as part of AGL release
- AGL environment set up provides extra features that may be enabled by device creators
- Device profiles (e.g., Telematics, ADAS) will be provided in AGL Extra Features
- Yocto layer – meta-agl-extra
• Place for developing code that may eventually make it into AGL Core or Extra Features
• Snap shot builds for experimental features to facilitate collaboration
• Community BSPs without official support
• Snap shot builds may be provided for Community BSPs
• No formal QA – basically whatever the community can provide
• Defined by Yocto layer – meta-agl-devel
Software Configuration Requirements

- AGL Demonstrator Code
- AGL Community Development
- AGL Extra Features
- AGL Core Distribution

Environment for demonstrator and new feature development
AGL Demonstrator Code

- Code developed to demonstrate specific features and/or releases of AGL
- CES 2017
- Automotive Linux Summit 2016
- Intended for “one shot” development
- Provided “as-is”
- Yocto layer – meta-agl-demo
Release Management

- Twice per year release of AGL Distribution includes:
  - AGL Core Distribution and Extra Features
  - All code and tooling with test results
  - Full test results for reference BSPs
  - As-Is demo code, Community Developed features, and BSPs
- Support biannual releases with code fixes for six months
- Long term support (2+ years) for selected releases
- Daily snapshot builds for specific configurations
- Pre-release candidates to allow developer collaboration and coordinated testing
AGL Yocto Layers

- AGL Demonstrator Code
- AGL Community Development
- AGL Extra Features
- AGL Core Distribution

AGL Test Framework

- meta-agl-demo
- meta-agl-devel
  - meta-agl-sota
  - meta-<BSP>
- meta-agl-extra
  - meta-ivi-common
  - meta-agl-security
- meta-agl
  - meta-agl
  - meta-agl-bsp
  - meta-ivi-common
  - meta-agl-security
- meta-poky
- meta-oe
- meta-<BSP>
Get The Code

• Pre-built binaries and source tar balls available
  • https://www.automotivelinux.org/software/download

• Latest Source Code and Build Instructions
  • https://wiki.automotivelinux.org/agl-distro/source-code
Build Options

• Once you have the repos set up use
  $ source meta-agl/scripts/aglsetup.sh –h
• To determine available boards and build options
• Example – Build QEMU AGL Demo
  $ source meta-agl/scripts/aglsetup.sh -m qemux86-64 agl-demo agl-netboot agl-appfw-smack

  $ bitbake agl-demo-platform
## Summary

<table>
<thead>
<tr>
<th>Source Location</th>
<th>Layer</th>
<th>QA Performed</th>
<th>Release Support</th>
<th>Daily Build and CI Builds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staging (or remote)</td>
<td>Meta-agl-demo</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Staging (or remote)</td>
<td>Meta-agl-devel</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Src (or remote)</td>
<td>Meta-agl-extra</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Src (or remote)</td>
<td>Meta-agl</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

- **Examples**
  - ALS and CES Demo apps belong in meta-agl-demo
  - Meta-agl-sota belongs in meta-agl-devel
  - Meta-iot-appfw belongs in meta-agl-extra
Getting Involved with AGL
Getting Involved

• AGL Wiki
  • https://wiki.automotivelinux.org/
• Single sign-on for AGL sites including Jira, git, gerrit, DOORS NG, and the AGL Wiki
  • Uses LF Identity
• Mail list for technical discussions
  • https://lists.linuxfoundation.org/mailman/listinfo/automotive-discussions
• IRC for technical discussions
  • #automotive on freenode.net
Getting Involved

• Most subsystems in need of developers and maintainers particularly user space
• Application developers needed
• Weekly developer calls on Tuesdays at 13:00 UTC
  • Info at https://wiki.automotivelinux.org/dev-call-info
• Check Jira for open issues and tasks that need to be done
  • https://jira.automotivelinux.org/
Contribution Process

• Code development process is documented
  • [https://wiki.automotivelinux.org/agl-distro/contributing](https://wiki.automotivelinux.org/agl-distro/contributing)

• Process continues to evolve as we mature
Git and Gerrit

• AGL uses git for version control and gerrit for code reviews

• Code and patch submissions are via gerrit and use the gerrit review and merge process

• These can be found at
  • https://gerrit.automotivelinux.org
  • https://git.automotivelinux.org
Continuous Integration

• Using Jenkins for Continuous Integration
• Patches
  • All changes submitted to gerrit are built immediately by Jenkins.
  • Successful build gives +1 to new code in Gerrit
  • Build failure -1 in gerrit
• Daily Snapshot builds
  • Available for reference BSPs
  • May add community BSPs later this year
  • https://download.automotivelinux.org/AGL/snapshots/master/
Automated Test

• Fuego (LTSI Jenkins Test Automation) being integrated into process

• More information
  • https://wiki.automotivelinux.org/agl-testframework
Summary

- Brilliant Blowfish 2.0.1 patch release available
- BB 2.0.2 will be available Week 37
- Additional F2F Meetings being planned for Charming Chinook
- Daring Dab – July 2017
Schedule F2F Workshops

- Recommend at least every six weeks
- Jul 12 - 14 Tokyo (ALS)
- Sep 7 – 8 Munich (AGL AMM)
- Oct 14 - Berlin or nearby (after ELC-E)
- Nov 17 – 18 – Yokohama (Finalized)
- Dec 15 – 17 Japan, TBD (Final CES integration)
- Ship demo to Las Vegas to arrive by Jan 3, 2017
Q&A

Tweet questions to @VStarWalt
THANK YOU