Building a Remote Control Robot with Automotive Grade Linux

Leon Anavi
Konsulko Group
leon.anavi@konsulko.com
Embedded Linux Conference Europe 2017
23-25 October, Prague, Czech Republic
Agenda

- Overview of AGL
- Selecting hardware components and building a simple remote controller robot
- Integrating and contributing additional software to the upstream of AGL
- Conclusions
- Q&A
Could other industries benefit from the features of Automotive Grade Linux (AGL)?
Requirements

Common requirements for an embedded Linux distribution for Internet of Things (IoT):

- **Build system and a development toolchain**
- **Security**
- **Over the air software updates**
- **Graphics and audio stack**
Automotive Grade Linux

- Project of the Linux Foundation
- In-Vehicle-Infotainment (IVI) GNU/Linux distribution
- Based on the Yocto Project and OpenEmbedded
- Founded in 2014
AGL Members

Platinum

**DENSO**

**Mazda**

**Panasonic**

**Renesas**

**SUZUKI**

**TOYOTA**

Gold

**HONDA**

The Power of Dreams

**NTT DATA**

NTT DATA MSE Corporation

Silver

**AISIN AW CO.,LTD.**

**Continental**

**Mercedes-Benz**

**DRIMAES**

**FUJITSU TEN**

**iridet**

**Mitsubishi Electric**

**NISSAN**

**Pioneer**

**QUALCOMM**

**WIND**
AGL Releases

- Funky Flounder 6.0.0 - scheduled for Jul 2108
- Electric Eel 5.0.0 - scheduled for Dec 2017
- Daring Dab 4.0.0 - Jul 2017
- Charming Chinook 3.0.0 - Jan 2017
- Brilliant Blowfish 2.0.0 - Jul 2016
- Agile Albacore 1.0 - Jan 2016
- AGL Unified Code Base (UCB) - 4 Jan 2016
AGL Core Technologies

- Qt/QML HMI
- HTML5
- GStreamer
- Weston
- Wayland
- SOTA Client & OSTree
- DBus
- systemd
- Linux kernel
- Security
- AppFW, Cynara, SMACK
AGL Yocto/OE Layers

- poky
- meta-agl
- meta-agl-demo
- meta-agl-devel
- meta-agl-extra
- meta-intel-iot-security
- meta-oic
- meta-qt5
- meta-updater
AGL Supported Devices

- Renesas Gen3 and Gen2 boards
- Intel Joule & Minnowboard
- TI DRA7xx EVM (Vayu)
- Raspberry Pi 2/3
- Dragonboard 610-c
- i.MX6 SABRE & HummingBoard
AGL Developer Tools

- **Git & Repo**
- **Gerrit** [https://gerrit.automotivelinux.org/](https://gerrit.automotivelinux.org/)
- **Jenkins** [https://jenkins-new.automotivelinux.org/](https://jenkins-new.automotivelinux.org/)
- **JIRA** [https://jira.automotivelinux.org/](https://jira.automotivelinux.org/)
- **Wiki** [https://wiki.automotivelinux.org/](https://wiki.automotivelinux.org/)
- **New documentation site** [http://docs.automotivelinux.org/](http://docs.automotivelinux.org/)
Building a RC Robot

Required hardware:

- Single board computer
- Chassis and DC motors
- Motor driver
- Sensors
- Batteries
Why Raspberry Pi 3?

- Low cost credit-card-sized computer
- Good software support in AGL
- Variety of add-ons
- Huge community
Some DC motor controllers for Raspberry Pi to consider:

- **ZeroBorg (TI DVR 8833)**
- **Picon Zero (TI DRV8833)**
- **Explorer pHAT (TI DVR8833)**
- **RasPi Robot Board (TB6612FNG)**
- **RTk.RPi.MCB (SN754410)**
DIY Motor Driver Board

Half-H motor drivers:

- **TI DVR8833**
- **TI L293D**
- **TI SN754410**
Controlling the Motors

- Software emulated PWM
- WiringPi library
- 2 DC motors controlled by SN754410 and powered by 4 AA batteries

```c
// Set pin mode
pinMode(motor1pin1, OUTPUT);
pinMode(motor1pin2, OUTPUT);
pinMode(motor2pin1, OUTPUT);
pinMode(motor2pin2, OUTPUT);

// Software PWM
enablePWM(motor1pin1, speedMax);
enablePWM(motor1pin2, speedMax);
enablePWM(motor2pin1, speedMax);
enablePWM(motor2pin2, speedMax);

void forward(int speed)
{
    softPwmWrite(motor1pin1, speed);
    softPwmWrite(motor2pin1, speed);
    motor1(HIGH, LOW);
    motor2(HIGH, LOW);
}
```
Sensors

- Ultrasonic sensor (HC-SR04)
- IR line tracking (TCRT5000)
- Triple Axis Compass Magnetometer (HMC5883L)
- Other I2C sensors for measuring temperature, humidity, colors, light, etc.
- Raspberry Pi Camera module V2
Communication

Built-in features in Raspberry Pi 3:

- WiFi
- Ethernet
- Bluetooth Low Energy (BLE)

Extended capabilities through add-ons:

- 433MHz radio transmission
- Infrared receiver
Example: IR Receiver

Hardware:
- **TSOP34838 Infrared Receiver**

Software:
- **LIRC (Linux Infrared Remote Control)**
Putting Things Together

UART

IR

I2C

HC-SR04

Motors
Building AGL

- **Get the source code:**
  
  ```
  repo init -u https://gerrit.automotivelinux.org/gerrit/AGL/AGL-repo
  repo sync
  ```

- **Setup the build environment:**
  
  ```
  source meta-agl/scripts/aglsetup.sh -m raspberrypi3 agl-demo agl-netboot agl-appfw-smack
  ```

- **Build an image:**
  
  ```
  bitbake agl-image-minimal
  ```
AGL Images

Common AGL targets are:

- agl-image-minimal
- agl-image-ivi
- agl-demo-platform
Customizing the Image

The quickest way to modify AGL image is to:

- Add new Yocto/OE layers to `conf/bblayers.conf`
- Expand the image through `conf/local.conf`, for example:

```
INSTALL_image_append = "lirc"
```
Contributing to AGL upstream

- Report an issue or a new feature in JIRA
- Modify the source code
- Include references to the JIRA issue in the Git commit messages
- Contribute to the upstream following the AGL Gerrit workflow
What is AGL Gerrit?

- Gerrit is web-based team code collaboration tool for code reviews of Git repositories
- Gerrit is free and open source software written in Java and available under Apache License v2
- Create an account for AGL Gerrit at identity.linuxfoundation.org and get started
AGL Gerrit Workflow

- Reviewer
- Repo
- Pending changes under review
- AGL repo
- Submit
- Pull
- Push
- Approve
- Fetch
AGL Communication Channels

- **AGL mailing list**
  
  https://lists.linuxfoundation.org/mailman/listinfo/automotive-discussions

- **Weekly Developer Call**
  
  *(Tuesday 14:00 - 15:00 UTC)*

  https://wiki.automotivelpinux.org/dev-call-info

- **IRC**
  
  channel #automotive on freenode.net
Conclusions

- Open source is compressing the development cycle for a faster route to the market
- AGL is based on a top of already proven open source software technologies
- AGL is entirely open source project that offers an open source software stack useful not only to the automotive industry but also to various Internet of Things (IoT)
What’s next?

- Designing using KiCAD an open source hardware Raspberry Pi add-on board for controlling the RC robot

- Releasing a stable final version of AGL Electric Eel 5.0.0 by the end of 2017

- Starting AGL Funky Flounder 6.0.0 at the beginning of 2018
Thank You!

Useful links:

- http://docs.automotivelinux.org/
- https://wiki.automotivelinux.org/start
- https://wiki.automotivelinux.org/agl-distro/agl-raspberrypi
- https://github.com/leon-anavi/rpi-examples