IoTivity on TIZEN
How To?

OpenIoT, San Diego USA <2016-04-04>
https://wiki.iotivity.org/tizen

Philippe Coval
Samsung Open Source Group / SRUK
philippe.coval@osg.samsung.com
Who Am I?

- Philippe Coval
  - Software Engineer from Samsung OSG
    - belongs to SRUK team, based in Rennes France
  - Tizen co-maintainer and IoTivity contributor
  - Interests:
    - Libre Soft/Hard/Ware, Communities, Interoperability
  - Ask me online for help or resources:
Agenda

• Introduction
  – Where do we go? When?
  – Where from?
  – Why Tizen and IoTivity?

• How to get it?
  – In Tizen platform(s)
  – On current Tizen products
  – And beyond
Once upon a time...
The vision to 2020

• Samsung's CEO Boo-Keun Yoon at CES2016:
  - "And five years from now, every single piece of Samsung hardware will be an IoT device, whether it is an air purifier or an oven."
  - "Without this kind of openness, there won't be an Internet of Things because the things will not fit together"
Tizen ecosystem

- An open source software platform
  - Announced in 2011 as LinuxFoundation collaborative project
- Shipped into consumer electronic products
  - 2013: Camera (NX1, NX300 …)
  - 2014: Wearables : Gear2, GearS
  - 2015: Mobile Samsung Z1 (Mobile 2.3), store
  - 2015: TVs (J serie), Z3 (Mobile 2.4)
  - 2016: And more to come
    - Refrigerator "Familly Hub", SmartHome, ConnectAuto...
Tizen an open platform

• GNU/Linux distribution that uses
  – mainline Linux Kernel (3.4, 3.10, 3.14, 4.1...)
  – uses RPM packages which are built using GBS

• Introduces a profile concept:
  – allows companies to standardize on a single base,
  – so every new product is not running a new OS

• Provides an application framework
  – Native or Webapps
IoTivity connects devices

- Seamless **device-to-device** connectivity for IoT
  - Discovery, Connection, Transmission, Security, Data & Device Management
- **C & C++** shared library
  - FLOSS: Apache-2.0 (hosted by Linux Foundation)
- Industry support:
  - Open Connectivity Foundation (OCF)
    - ex- Open Interconnect Consortium (OIC)
    - Samsung, Intel, Cisco, GE, Qualcomm, Electrolux…
IoTivity development model

- **Cross Platform:**
  - Linux, Android, Tizen, Arduino, OSX…

- **Uses** `scons` **build system**
  - with various configuration options:
    - log, security, transport (IP, WiFi, BT, BLE)

- Ships a couple of examples

- **Continuous integration**
  - Build iotivity using: autobuild.sh
Tizen is supported by IoTivity

- Tizen is one of supported targets:
  - autobuild.sh tizen
    - Uses GBS to produce RPMs for platform
- Tizen target is enhanced Linux target
  - mostly for controlling network adapters and log output (dlog)
  - with minimal specific code (ifdef)
    - grep -r __TIZEN__ * | wc -l # only 31
- build bots connected to jenkins and gerrit
Is the job done?

- Not yet, let me explain why
  - There is no single Tizen
  - Platform(s) != Product(s)
- But everything is (almost) ready to be supported
  - at platform level
  - or as standalone application
  - and beyond
There is no Tizen's maze

- But **many** codebases
  - Tizen = ( Products / profiles ) * platform * versions

- 3 branches:
  - 1.x to validate technology on reference devices
  - 2.x shipped into current products
  - 3.x as R&D platform, split into **profiles**

- Today, 2 & 3 are still evolving in parallel
Tizen 3 Profiles

• Open governance on http://tizen.org
• Tizen:Common rules them all
  – It's not a profile
  – but can be used as a base for profiles
• Other Tizen 3 profiles are derived from Common
  – ie: 90% of Tizen:IVI (automotive) is Tizen:Common
  – Wearable + Mobile + TV
  – Micro for IoT
Tizen:Common

- Regular GNU/Linux distro (RPM based)
  - Min features: Security, Graphics, Comms, Browser, AppFW
- for general purpose development hardware
  - Intel/ARM, 32/64bits CPUs, on PC, VM, SBC, OSHW
- Open upstream development to any
  - and/or work with supported “Contrib repository”
- Platform developers use those tools:
  - git, GBS, gerrit, OBS, rpm, zypper
IoTivity from Tizen:Common

- Just Install it on Tizen:Common
  
  ```
  zypper ar \
  http://download.tizen.org/live/Contrib:/Common/\x86_64-wayland/Contrib:Common.repo
  zypper in iotivity-devel
  ```
  - Since 0.9 (2015-01) as community contrib
  - Then maintained by project in platform/upstream

- So package spreaded to Tizen:IVI, TV...
- Just rebuild it using GBS...
Git Build System (GBS)

- Tizen supported build system
  - to produce RPMs (per profile / per version)
  - Work along OBS RPM repos like zypper
- Install tizen tools for your favourite GNU/Linux Distro
  - configure ~/.gbs.conf (hint: git clone tizen-helper)
- Need file: `packaging/iotivity.spec`
  - hint: inspire for upstream one in tools/tizen
- `gbs build -P tizen_common -arch x86_64`
Showcase #1

• Server shares resource
  – LED from MinnowBoard
  – Tizen:Common + mraa
• Client changes state
  – From Max (Calimari Lure)
• Observer (FRI2 Tizen Yocto)
  – Sends SMS
    • Using ofono
• Phone (Samsung Z1)
  – receives sms
    • Using Tizen CAPI
IoTivity on Tizen Mobile
To Tizen:2.4:Mobile platform

- From Tizen:Common to Tizen:2.4:Mobile
  - g++ 4.9.2 to armv7l supported by IoTivity
  - Rebuild dependencies if needed using gbs
    - scons, boost (1.57+) + uteemper, dos2unix, boost-jam
- It just works on TM1 Reference device
  - deploy as root: sudo rpm -i iotivity*.rpm
  - so we validated kernel multicast support
- But there it stops here: TM1 is not a product!
What's a Tizen product?

- Built on FLOSS
  - http://opensource.samsung.com
- Belongs to some profile:
  - Mobile, Wearable, TV…
- Can install applications from Tizen Store
- **SDK** for 3\textsuperscript{rd} party developers (IDE or CLI)
  - WebApps: High level Javascript API (W3C, Jqm, TAU…)
  - Native Apps: C/C++ APIs, **EFL** (C) for UI/UX, MT…
IoTivity on Samsung Z1

- Z1 is also supporting Tizen:2.4
- How to make an app that link with iotivity library
  - Using Tizen-SDK create a native app (EFL)
  - generated skeleton: EFL main loop, with basic UI
  - import libs (just unpack iotivity*.rpm, update eclipse's .cproject)
    - add "/${ProjName}/usr/include" "/${ProjName}/usr/lib"
    - Link to oc, oc_logger, octbstack, uuid, connectivity_abstraction
  - lunch IoTivity in a thread and proceed callback functions
- Deploy .tpk to device and observe trace with sdb dlog
  - Mission accomplished, polish for tizenstore
Showcase #2:

- **IoTivity server running on Tizen:IVI**
  - Share resource: LED from minnowboard Max
  - Has a button (lure) to change state

- **IoTivity client app running on Tizen mobile Z1**
  - Changes state of resource
  - Observes resource state
    - when changed from anywhere

- [https://vimeo.com/161074400#iotivity-tizen-z1-gears-20160331](https://vimeo.com/161074400#iotivity-tizen-z1-gears-20160331)
Showcase: IoTivity Mobile + Wearable

"IoTivity Tizen HowTo"
on MinnowMax (Tizen:IVI)
Samsung Z1 (Tizen:2.4:Mobile)
Gear S (Tizen:2.3:Wearable)

https://wiki.iotivity.org/tizen
CC BY-SA @TizenHelper @SamsungOSG
The Yocto project's option
The Yocto Project

- Linux foundation collaborative project
  - to create custom Linux-based systems
  - for embedded devices
- It uses OpenEmbedded framework
- Industry & community support:
  - point of convergence?
- Note: Automotive distros are based on it
  - GENIVI, LFAGL, Tizen:IVI (Yocto)
Tizen: 3.0 Yocto

- Alternate build system
  - for building Tizen images for different arch
  - using the tools provided by the Yocto Project

- Benefit: standalone & extensible
  - S-OSG provided support for RaspberryPi2 (1 and 0 too)
  - Many hardware support (cheap SBC starting at 5 USD)

- Micro profile (headless) perfect for IoT

- Note: Yocto and GBS projects are not interoperable
IoTivity supports Yocto

- Through **meta-oic layer**
  - BBLAYERS += ".../meta-oic"
  - provides iotivity **recipe** (1.0.0+)
  - Patches kernel with .config fragments (if using linux-yocto)
- **meta-yocto-demo image** with iotivity-example
- Successfully tested on:
  - Tizen:Common, RaspberryPI 1, RPI 2, FRI2...
  - GENIVI Demo Platform on MinnowMax + calimari lure
  - LFAGL on MinnowBoard Max
Showcase #3 : IoTivity Yocto + TM1

- IoTivity deployed on:
  - Tizen DIY Fan
    - Tizen:Yocto on RaspberryPI 1
    - Relay on GPIO (3.3 V) + NPN
  - Controled by TM1 (Tizen:2.4 Mobile)
  - Controled by LFAGL's HomeScreen
    - Running on MinnowMax
      - (AGL/yocto)
    - On OSVehicle
      - from #CampOSV Rennes France
  - https://vimeo.com/156307187#iotivity-agl-demo-platform-20160222rzr
Showcase #3: IoTivity/Yocto + TM1

"IoTivity Tizen Fan"
controlled by
Automotive Grade Linux
and TM1 on OSVehicle

https://wiki.iotivity.org/community
CC BY-SA @TizenHelper @SamsungOSG
Want more ?
Tizen Wearable

- Gear 2: BTLE no WiFi but webapp only supported
  - Need to be paired to Android phone
    - IoTivity bundled into APK: Communication LAN (WiFi)
    - Support Samsung Accessory Protocol (SAP)
      - Forward events from/to Tizen device

- Gear S: like Gear2 with WiFi
  - Can lunch a unix process from developer mode (sdb shell)

- Gear S2 supports Native applications
  - Same as Z1 Tizen:2.4:Mobile => Adapt to Tizen:2.3.1:Wearable?
Almost all covered

- TV: Web APIs are supported
  - Apply partner program for Native Applications
  - Use NACL support? (alt: WebAssembly or asm.js?)
  - Ask for demo on Tizen TV at OpenIot IoTivity’s booth
- Products based on Tizen but not branded as
  - Camera, White goods?
  - WebSockets to gateway with IoTivity support
- Your Tizen devices
  - Rebuild Tizen:3.0 for (un)supported arch (Yocto/GBS)
  - IoTivity is supported by both build system
  - Adapt and share feedback about DIY or certified devices
Beyond Tizen's ecosystem too

- Tizen IoTified devices to talk with:
  - GNU/Linux systems (Debian, etc)
  - Yocto/Poky based OS (LF AGL, GENIVI, Ostro-OS)
  - Android phones (and other devices)
  - and others OS: OSX, iOS, Windows?
  - Microcontrollers (MCU like Arduino)
  - Other RTOS to be supported by IoTivity

- Think about gateways to other specific protocols too
  - Automotive, SmartHome (SmartThings), Smart Cities (LPWAN)
Next steps

- Check about other transport:
  - BT, BLE & Plugins (ideas: Sigfox on Artik, LoRa, ...)
  - Cloud, Gateways/Bridges
- WebApps/Javascript APIs:
  - Tizen hybrid applications or services
    - Web UI + Message port
  - Iotivity-node
- UI/UX
  - Discovery & Visualisation, establish behaviours...
Summary

- **Openness** is needed for IoT
  - Tizen and IoTivity are open
- Tizen has many **profiles**
  - Don't be confused between platform and products
- IoTivity is supported by:
  - Tizen 3.0 platform
  - Tizen 2.3+ products as native application(s)
  - Other OS (CPU and MCU)
Annexes
References

- **Entry points:**
  - https://OpenIoElc2016.sched.org/event/6DBC/
  - https://wiki.iotivity.org/tizen
  - https://wiki.iotivity.org/community

- **Mentioned:**
  - https://wiki.tizen.org/wiki/Yocto
  - https://wiki.iotivity.org/yocto
  - https://youtu.be/R9UHnrP_tRQ# (Samsung CES 2015)

- **Related:**
  - http://www.slideshare.net/SamsungOSG
  - https://fosdem.org/2016/schedule/event/connected_tizen/
  - https://wiki.iotivity.org/compiler_support
  - https://at.projects.genivi.org/jira/projects/GOCF
  - https://jira.automotivelinux.org/browse/SPEC-158
Annex : Iotivity Tizen support

- If no RPM available : just rebuild from tizen's contrib repo
- Tizen:3.0:Common : supported (OBS) (since 0.9, now 1.0.1)
  - Same for Tizen Yocto : meta-oic support (1.0.0 now 1.0.1)
- Tizen:2.4:Mobile + Tizen:2.3:Mobile
  - as native app 1.0.1+ (Tested on TM1 / Z1)
- Tizen:2.3.1:Wearable (as native app?) (WIP)
- Tizen:2.3:Wearable (aka 2.2.1.3+)
  - as developer process (Tested on GearS)
- Tizen 1.0 to 2.2 (Mobile) : g++-4.5 < g++-4.6
  - TODO : backport code or just use C SDK ?
Annex : Demo Sources (WIP)

- Staging area for demos @ http://github.com/tizenteam/
- iotivity:
  - contrib branch for tizen or tizenteam
- iotivity-example:
  - Minimal client/server/observer + mraa + integration files
- meta-yocto-demos:
  - Uses meta-oic
  - To stage changes, recipes and configurations
  - Helper to build images with simpler “make” call
  - Request your SBC to be supported
Thanks

Samsung, LinuxFoundation, OCF/OIC, City of San Diego, Intel
Tizen, IoTivity, EFL, Yocto, GENIVI, LF AGL, contributors
Artik. RPI/Broadcom, OSHW, FLOSS, communities,
freenode #tizen, TizenTeam, TizenExperts
Flaticons (CC BY 2.0), libreoffice
ELC2016/OpenIoT's team,
You!