Evolution to OS of Everything

Tizen Micro profile for low-end IoT devices

趙庸鎮
(Cho, Yong Jin)
drajin.cho@samsung.com

Software Center, Samsung Elec.
Contents

Part I. OS of Everything in IoT
Part II. Tizen Micro Profile
Part III. PoC of Tizen Micro Profile
Part IV. Future Plans
Part I,
OS of Everything in IoT
Internet of Things

Internet uses Things

Things use Internet

Part I. OS of Everything in IoT
Internet of Things

Q) Why do you want to use Internet?

A) If I know when my master is coming, I can make room temperature more comfortable for my master. It will make my master happier.
Q) How do you know when he is coming?

A) It depends.
My master is usually coming home by his car.
I may guess his arrival time,
if I know when he is parking his car at home.
Internet of Things

Q) If your master goes out after parking, ...

A) In the estimated time, if my master isn’t coming home, I will send message to him.
   “Air-conditioner: I started to make room temperature.”
   He can stop me or let me know his arrival by replying message.
   “Master: I will be home at 4:00pm.”
Internet of Things

The more utilizations of Internet, the smarter services by IoT devices

Things use Internet
What’s needed in the things?

Open platform, allowing to add new IoT service app which provides more personalized service and make the device much smarter

**Application framework / API**

Utilize the information from Internet of Things
Provide smart service and add more value to device
- Personalized service
- Context-aware service

Part I. OS of Everything in IoT
Another way?

Device provides only the pre-defined interface to the connected. Service app is running on the other device or cloud which can provide the more personalized service.
Part I. OS of Everything in IoT

OS of Everything in IoT, **Tizen**

Part I. OS of Everything in IoT

[w/o AppFW]  [AppFW]

Internet of Things

Nano  Micro  Full (TV, Mobile)
Part II,
Tizen Micro Profile
Tizen Micro Profile

Tizen profile for developing IoT device which has application framework and exposes APIs, that allows to add IoT services.

Utilize the information from Internet of Things
Provide smart service and add more value to device
- Personalized service
- Context-aware service
Requirement - Market

Model w/o IoT

Model with IoT

Physical Integration
UART, I²C, ...

Cost Effectiveness
32M FLASH, 64M RAM
Requirement - Apps

Tizen Micro Apps want to use
- Internet service from cloud
- Service from the other connected device
- Local device service

Tizen Micro Apps need to provide
- Internet service to cloud
- Service to the other connected device
- Event handler from local device

use service from IoT + provide service to IoT
Architecture

Web of Things
connects things through the existing Web technology
(XML/JSON over HTTP)
Architecture – Web of Things

JavaScript is the most popular language in web, not only client-side but also server-side.
Architecture – Device BUS
Architecture – Device BUS

Extends to the external device

- Service App
- Device App
- Web Server
- JavaScript App FW
- Device BUS
- Device

ZigBee, Z-Wave, ...
Tizen Nano Device

- Tizen Companion Device
- Device App installed on Tizen Device
- Equivalent Device Function of Tizen Micro, added to Tizen Device
Web of Things and Device BUS features in Tizen Micro profile can be integrated into the existing Tizen profiles.
Tizen 3.0 & Micro Profile

Tizen GIT repository

git://review.tizen.org

/platform/framework/native
/platform/framework/web
/platform/kernel/linux-3.10
/platform/upstream/glib
/platform/upstream/nodejs
/platform/profile/mobile
/platform/profile/tv
/platform/profile/micro

Tizen Mobile Profile

Tizen TV Profile

Tizen Micro Profile
Part III, PoC of Tizen Micro Profile
Proof of Concept

- Memory footprints: 32M FLASH, 64M RAM
- Web of Things with JavaScript framework
- IoT scenarios

<table>
<thead>
<tr>
<th>Framework</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Nginx</td>
<td>Node.js</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Core Libraries</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>glib</td>
<td>busybox</td>
</tr>
<tr>
<td>openssl</td>
<td>...</td>
</tr>
</tbody>
</table>

Linux Kernel

20M FLASH
40M RAM
(Remote UI scenario)
Demo Scenario

Part III. PoC of Tizen Micro Profile

<table>
<thead>
<tr>
<th>&lt;&lt;service&gt;&gt;</th>
<th>Auto Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tizen Micro</td>
<td>Audio</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;&lt;service&gt;&gt;</th>
<th>Temp Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tizen 3.0 TV</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;&lt;device&gt;&gt;</th>
<th>Person In/Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARTIK10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;&lt;device&gt;&gt;</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;&lt;service&gt;&gt;</th>
<th>22℃</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUT</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;&lt;service&gt;&gt;</th>
<th>Auto Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tizen Micro</td>
<td>Air-conditioner</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;&lt;service&gt;&gt;</th>
<th>Personal News</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tizen Micro</td>
<td>Printer</td>
</tr>
</tbody>
</table>

Temperature: 22℃
Part IV,
Future Plans
Future Plans

[Release 1. ’15. 4Q]
step 1. building Tizen Linux kernel and system on Raspberry Pi2 using Yocto
step 2. integrating Nginx and Node.js
step 3. building Device BUS

[Release 2. ‘16. 1Q]
- footprint optimization: 32M FLASH, 64M RAM
- product-line management

[Release 3. ‘16. 4Q]
- additional features: remote access, multimedia, etc.
Tizen Micro profile on RTOS

Samsung opened development of IoT.js, JavaScript engine and application framework for low-end IoT devices.

- Node.js
- V8
- Linux
- IoT.js
- JerryScript
- RTOS

- iotjs.net
- jerryscript.net

Part IV. Future Plans
The best way to connect everything

Part IV. Future Plans

Inter-operability

Easier integration and spread of IoT

Full

Micro

Nano

OS of everything in IoT

OIC

WoT

TIZEN 开发者大会 2015 中国 深圳
Q & A
Tizen Nano Device

Tizen companion device, which adds device function into another Tizen device with installing device app on it.

Provide the pre-defined interface

HUB

Tizen Device

Equivalent to Tizen Micro
# Packages Used in PoC

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Git Repository</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux-3.10</td>
<td>platform/kernel</td>
<td></td>
</tr>
<tr>
<td>openssl-1.0.2</td>
<td>platform/upstream/openssl</td>
<td>1.0.1</td>
</tr>
<tr>
<td>nodejs-0.12.5</td>
<td>platform/upstream/nodejs</td>
<td>0.12.0</td>
</tr>
<tr>
<td>nginx-1.6.2</td>
<td>platform/upstream/nginx</td>
<td>[new]</td>
</tr>
<tr>
<td>libglib2-2.42.0</td>
<td>platform/upstream/glib</td>
<td></td>
</tr>
<tr>
<td>busybox-1.23.1</td>
<td>platform/upstream/busybox</td>
<td>1.22.1</td>
</tr>
<tr>
<td>gettext-0.19.4</td>
<td>platform/upstream/gettext</td>
<td>0.18.3.2</td>
</tr>
<tr>
<td>pcre-8.36</td>
<td>platform/upstream/pcre</td>
<td>8.31</td>
</tr>
<tr>
<td>libffi-3.1</td>
<td>platform/upstream/libffi</td>
<td></td>
</tr>
<tr>
<td>zlib-1.2.8</td>
<td>platform/upstream/zlib</td>
<td></td>
</tr>
<tr>
<td>uclibc-0.9.33.2</td>
<td>platform/upstream/uclibc</td>
<td>[new]</td>
</tr>
</tbody>
</table>
Internet of Things

Q) What else do you want more?

A) It will be helpful to know
   - how the temperature is in the car
   - whether he is excising or not
   - where he is exactly at home
   - how the weather or the forecast is
   …

I can provide more personalized service.
Demo Scenario(1) person in

Part III. PoC of Tizen Micro Profile

Tizen Micro Audio

Auto Play

Tizen Micro Printer

Tizen 3.0 TV

ARTIK10

Temp
Change

Person
In/Out

Temp

Person

Auto Control

Tizen Micro Air-conditioner

22℃

IN

22℃

<<service>>

Temp

Person

In/Out

<<service>>

Person

<<service>>

Temp

22℃

<<device>>

22℃

<<device>>
Demo Scenario(2) **new service added**

**Part III. PoC of Tizen Micro Profile**

- **Auto Play**
- **Tizen Micro Audio**

- **Personal News**
- **Tizen Micro Printer**

- **Tizen 3.0 TV**
  - **ARTIK10**
  - **Temp**
  - **In/Out**
  - **Person**

- **Auto Control**
- **Tizen Micro Air-conditioner**

- **Temp Change**
- **Person**
- **Temp**
- **Person**

- **22℃ IN**
Demo Scenario(3) temperature up

Part III. PoC of Tizen Micro Profile

- Tizen Micro Audio: Auto Play
- Tizen Micro Printer: Personal News

Diagram:
- Tizen 3.0 TV
- ARTIK10
- Auto Play
- Personal News
- Auto Control
- Air-conditioner

Table:

<table>
<thead>
<tr>
<th>Service</th>
<th>Device</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temp</td>
<td>Person</td>
<td>In/Out</td>
</tr>
<tr>
<td>Change</td>
<td>Temp</td>
<td>Person</td>
</tr>
<tr>
<td>25℃</td>
<td>23℃</td>
<td>IN</td>
</tr>
</tbody>
</table>
Demo Scenario(4) **person out**

Part III. PoC of Tizen Micro Profile

- **Auto Play**
  - Tizen Micro Audio

- **Personal News**
  - Tizen Micro Printer

- **Temp Change**
  - **Person In/Out**
  - **Temp**
  - **Person**

- **Tizen 3.0 TV**
  - **ARTIK10**

- **24℃**
  - OUT

- **22℃**

- **Auto Control**
  - Tizen Micro Air-conditioner
Demo Scenario(5) scheduled

Part III. PoC of Tizen Micro Profile

- **Auto Play**
  - Tizen Micro Audio

- **Personal News**
  - Tizen Micro Printer

- **Temp Change**
  - Person In/Out
  - Temp
  - Person

- **Tizen 3.0 TV**
  - ARTIK10

- **Temp**
  - 24℃
  - OUT

- **Temp**
  - 22℃

- **AM 6:00**

- **Auto Control**
  - Tizen Micro Air-conditioner