Build your smart device with Tizen-micro by using yocto in only one day

Biao Lu
Austin Zhang
Agenda

• Current states & existing Issues
• Tizen-micro Features
• Build your device quickly & flexibly
• Case study
• Summary
• Q & A
Current states

Images source:
http://gadgetsin.com/smartthings-make-your-home-smarter.htm
http://www.cnn.com/2014/03/28/tech/innovation/smart-device-communication/
Existing Issues

- Setup development environment is complicated.
- Whole stack knowledge is required.
- Various system components stability is hard to verify.
- Not easy to meet Footprint requirement.
How Can Tizen-micro help resolve these issues?
Tizen-micro 101

What is it?

Tizen-micro

• **One Tizen system with below features:**
  • Lightweight (headless) & Small footprint & Multi-tasking
  • Highly configurable & Flexibility
  • Stability & Verification
  • Clear & Quickly build
## Tizen-micro Architecture

<table>
<thead>
<tr>
<th>Customization</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Profile</strong></td>
<td></td>
</tr>
<tr>
<td>SmartHub</td>
<td>Samba</td>
</tr>
<tr>
<td>hostapd</td>
<td>Dnsmasq</td>
</tr>
<tr>
<td>SmartCamera</td>
<td>Face-integrated</td>
</tr>
<tr>
<td>encoder</td>
<td>decoder</td>
</tr>
<tr>
<td><strong>Common</strong></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>Python</td>
</tr>
<tr>
<td>nginx</td>
<td>Node.js</td>
</tr>
<tr>
<td>SQLite</td>
<td>libjpeg</td>
</tr>
<tr>
<td>Mosquitto</td>
<td>BlueZ</td>
</tr>
<tr>
<td>Multimedia</td>
<td></td>
</tr>
<tr>
<td>GStreamer</td>
<td>ALSA</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td></td>
</tr>
<tr>
<td>ppp</td>
<td></td>
</tr>
<tr>
<td>oFono</td>
<td></td>
</tr>
<tr>
<td>wpa_supplicant</td>
<td></td>
</tr>
<tr>
<td><strong>BSP</strong></td>
<td></td>
</tr>
<tr>
<td>Galileo</td>
<td>Minnow-Max</td>
</tr>
</tbody>
</table>

Linux Kernel

**Application**
- SmartHub
- Dnsmasq
- hostapd
- Samba

**SmartCamera**
- Face-integrated
- encoder
- decoder

**Common**
- Python
- Node.js
- nginx
- SQLite
- libjpeg
- Mosquitto
- BlueZ
- GStreamer
- ALSA

**Network**
- ppp
- oFono
- wpa_supplicant

**BSP**
- Galileo
- Minnow-Max
Tizen-micro Footprint

- SmartHub: 132M
- SmartCamera: 53M
- Base: 48M
- Network: 57M
- Multimedia: 79M
How to use Tizen-micro to build your device
Build your device quickly & flexibly

• Get Tizen-micro
  • Get all those codes/layers/pkg group definition/scripts … to build one Tizen-micro based system

• Tree structure glance

```
  ├── tizen-micro/
  │   ├── meta-micro-common/
  │   ├── meta-smarthub/
  │   ├── meta-intel/
  │       └── meta-galileo/
  │           └── meta-minnowmax/
  │
  │   └── setup/
  │       ├── setup.sh

```
Build your device quickly & flexibly (cont.)

- It used Yocto as build tools and some components repo
- The easiest way to create an image

```
$ ./setup.sh
$ bitbake micro-image
```

- Run in VM

```
# ./setup/vm-run.sh
```

Yocto project: https://www.yoctoproject.org/
Base system only

Base system only

Customization

Application

Profile

SmartHub

Samba

hostapd

DNSmasq

SmartCamera

Face-integrated

coder

decoder

OpenCV

Base

Python

Node.js

nginx

SQLite

libjpeg

Mosquitto

BlueZ

Common

Multimedia

GStreamer

ALSA

Network

ppp

neard

oFono

wpa_supplicant

Linux Kernel

BSP

Galileo

Minnow-Max

...
Build your device quickly & flexibly (cont.)

• Add the components as needed in previous image
  • Pre-defined middleware group

```bash
$ vi meta-micro-common/recipes-core/images/micro-image.bb
IMAGE_INSTALL += "packagegroup-multimedia"
```

• Your needed middleware components/groups

```bash
IMAGE_INSTALL += "your-middleware"
```

• Your application

```bash
$ vi your-app_0.1.bb
```
Add additional middleware and application
Build your device quickly & flexibly (cont.)

- Select BSP & Profile for previous image
  - BSP

```bash
$ ./setup.sh -b boardname
$ bitbake micro-image
```

- How to integrate your BSP codes into?

Put BSP under directory "meta-intel", e.g. meta-intel/meta-your-bsp

```bash
$ ./setup.sh -b your-bsp
```

- Profile

```bash
$ ./setup.sh -b boardname -p smarthub
$ bitbake smarthub-image
```
Add BSP / Profile support

Customization

Application ...

Profile

SmartHub

hostapd

Samba

Dnsmasq

SmartCamera

Face-integrated

encoder decoder

OpenCV ...

Common

Base

Python

Node.js

nginx

SQLite

libjpeg

Mosquitto

BlueZ

Multimedia

GStreamer

ALSA

Network

ppp

neard

oFono

wpa_supplicant ...

Linux Kernel

BSP

Galileo

Minnow-Max ...

TIZEN

DEVELOPER SUMMIT 2014 SHANGHAI
TIZEN开发者峰会（上海）
Build one smart device from scratch:
we can build our smart device in one very quick and workable way now!
Case study – Minnow-Max smart hub

Image from: http://www.minnowboard.org/meet-minnowboard-max/
Case study – Minnow-Max smart hub

- One stable and verified base system: Tizen-micro common base
- Additional product profile: smarthub
- Customized UI/UX
- One great HW platform: Minnow-Max
Case study – Minnow-Max smart hub (cont.)

• Create a BSP Layer

```plaintext
tizen-micro/
  .../
  meta-intel/
    meta-galileo/
    meta-minnowmax/
  ...
```

• New Layer config file (layer.conf)

```plaintext
BBPATH .= "${LAYERDIR}"

BBFILES += "${LAYERDIR}/recipes-*/*/*.bb \n           ${LAYERDIR}/recipes-*/*/.bbappend"

BBFILE_COLLECTIONS += "minnowmax"
BBFILE_PATTERN_minnowmax := "^${LAYERDIR}/"
BBFILE_PRIORITY_minnowmax = "5"
```
Case study – Minnow-Max smart hub (cont.)

- New Machine config file (machine/minnowmax.conf)

```
DEFAULTTUNE := "corei7-64"
require conf/machine/include/tune-corei7.inc
require conf/machine/include/x86-base.inc

PREFERRED_PROVIDER_virtual/kernel = "linux-minowmax"
PREFERRED_VERSION_linux-minowmax = "3.14%"

SERIAL_CONSOLES = "115200;ttyS0"
```
Case study – Minnow-Max smart hub (cont.)

- Kernel config file (linux-minnowmax_3.14.bb)

```plaintext
require recipes-kernel/linux/linux-yocto.inc

SRC_URI = "git://git://git.kernel.org...git;branch=linux-3.14.y;"
SRC_URI += "file://minnowmax.patch"
SRC_URI += "file://minnowmax.cfg"

LINUX_VERSION ?= "3.14.4"
LINUX_VERSION_EXTENSION ?= "-minnowmax"

SRCREV_machine = "62f236c734996f240d91daee2cb6a05669c7326c"

PR = "r0"
PV = "${LINUX_VERSION}"

COMPATIBLE_MACHINE = "minnowmax"
```
Case study – Minnow-Max smart hub (cont.)

• Build image

```bash
$ ./setup.sh -b minnowmax -p smarthub
$ bitbake smarthub-image
```
Last but not least 😊
Summary
Summary

- With the continuous innovation of smart devices, developers need easy way to enable base system for speeding up to put their ideas into reality.
- Tizen-micro offers many features for build your smart device in quickly way.
  - Full build environment
  - Stable and verified middleware
  - Small footprint
- Developer can quickly and flexibly build system through simple steps.
- Take Minnow-Max as an example to show how to build system using Tizen-micro.
Q & A
Image from:
www.toogoodcriticalfriend.co.uk/wp-content/uploads/2014/06/cv_57_7424481104157220867.jpg